Κωστόπουλος Δημήτριος
Μ.Π.Λ.Α.

LINEAR SPEEDUP

Εισαγωγή.

Αυτό το φυλλάδιο έχει στόχο να δείξει ακριβώς ότι εάν έχουμε κατασκευάσει μία μηχανή Turing M για να λύσει ένα πρόβλημα, να αποφασίσει μία γλώσσα και τρέχει σε χρόνο f(n), τότε μπορούμε να κατασκευάσουμε μία μηχανή Turing M' που να προσομοιώνει τη M σε χρόνο εf(n) + n + 2, όπου n είναι το μέγεθος της εισόδου.

Για να καταλάβουμε πώς μπορεί να κατασκευαστεί μία μηχανή ας δούμε ένα παράδειγμα. Έστω μία μηχανή Turing M που να κάνει πράξεις μεταξύ δύο δυαδικών αριθμών που έχουν πλήθος ψηφίων k. Τότε αυτή η μηχανή θα χρειαστεί περίπου k κινήσεις και περίπου 8 καταστάσεις για να εκτελέσει την πρόσθεση. 8 καταστάσεις για να προσθέσει τους 4 δυνατούς συνδυασμούς από σύμβολα, με ή χωρίς κρατούμενο. Υπάρχουν βέβαια και άλλες κινήσεις και καταστάσεις για τη μηχανή M, αλλά το πλήθος τους δεν εξαρτάται από το μέγεθος της εισόδου.

Πώς καταφέρουμε να φτιάξουμε μια γρηγορότερη μηχανή M'; Απλά, εκτελούμε τις πράξεις σε δεκαεξαδικό σύστημα. Ενώ θέλαμε k κινήσεις, θέλουμε k/4 κινήσεις γιατί η είσοδος θα μετατραπεί σε δεκαεξαδικό και θα έχουμε να προσθέσουμε k/4 ψηφία. Αρά μια μηχανή M' που κάνει μετατροπές σε από δυαδικό σε δεκαεξαδικό και μετά προσθέσεις σε δεκαεξαδικό θα είναι 4 φορές περίπου γρηγορότερη από τη M.

Όλο το ζουμί είναι να συμπιέσουμε την είσοδο. Αλλά επειδή τίποτα σε αυτή τη ζωή δε γίνεται χωρίς αντάλλαγμα, από κάπου θα χάνουμε. Κερδίζουμε σε χρόνο αλλά χάνουμε σε πολυπλοκότητα. Η M' χρειάζεται περίπου $\binom{16}{2}$= 120 καταστάσεις για τις πράξεις μεταξύ των δεκαεξαδικών ψηφίων, 240 αν λάβουμε υπόψιν μας και το κρατούμενο. Η μηχανή M' είναι πολύ πιο πολύπλοκη, έχει πιο μεγάλο πρόγραμμα και επομένως πιο δύσκολα υλοποιήσιμη. Ας δούμε όμως το σχετικό θεώρημα και την απόδειξή του.

Θεώρημα:

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

Απόδειξη:

Έστω M = (K, Σ, δ, s) μία k – string Turing μηχανή η οποία αποφασίζει την L και τρέχει σε χρόνο f(n). Θα κατασκευάσουμε μία k' – string Turing μηχανή M' = (K', Σ', δ', s') και η οποία θα τρέχει σε χρόνο εf(n) + n + 2, για ε > 0, με σκοπό να προσομοιώσει την M. Θα έχουμε k' = k για k > 1 και k' = 2 για k = 1. Πώς το επιτυγχάνουμε αυτό; «Απλά» κάνουμε τη μηχανή μας πιο πολύπλοκη κωδικοποιώντας πολλά σύμβολα σε ένα. Με τη συμπίεση των συμβόλων η μηχανή κάνει πιο γρήγορες πράξεις, αλλά το πρόγραμμα της μεγαλώνει δραματικά , γίνεται πολύπλοκο και πιο δυσνόητο.

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 Σ' = Σ U Σ^m.

Σε πρώτη φάση πρέπει να μετατρέψουμε την είσοδο της M έτσι ώστε η είσοδος της M’ να περιέχει μόνο m-συμπιεσμένα σύμβολα. Επειδή η είσοδος της M είναι και είσοδος της M’, η μετατροπή αυτή γίνεται στη δεύτερη ταινία της M’. Κάθε φορά που διαβάζουμε ένα σύμβολο από την είσοδο μεταφερόμαστε σε μία νέα κατάσταση η οποία θυμάται τι έχουμε διαβάσει μέχρι στιγμής. Κάθε φορά που διαβάζουμε m σύμβολα γράφουμε στη δεύτερη ταινία της M’ το νέο συμπιεσμένο σύμβολο. Εάν στο τέλος έχουν μείνει λιγότερα από m σύμβολα τότε συμπληρώνουμε με κενά. Αυτό γίνεται εύκολα προσθέτοντας Σ^i καταστάσεις με i = 1, 2, …, m.
Εάν τα σύμβολα είναι $\sigma_1, \sigma_2, ..., \sigma_{|\Sigma|}$ τότε θα έχουμε κάτι τέτοιο:
$(k, \sigma_i, ...)$ -> $(k_{\sigma i}, \sigma_i,$ ->, |>, ->, ...)
$(k_{\sigma i}, \sigma_j, ...)$ -> $(k_{\sigma i \sigma j}, \sigma_j,$ ->, |_|, -, ...)
…
$(k_{\sigma i\ \sigma j ... \sigma l}, \sigma_p …)$ -> $(k_{\sigma p}, \sigma_p,$ ->, $\sigma_{\sigma i\ \sigma j ... \sigma l}$, ->, ...)
με i, j, l, p ε {1, ..., |Σ|} και $\sigma_{\sigma i\ \sigma j ... \sigma l}$ το κωδικοποιημένο σύμβολο για τα συστοιχία συμβόλων σi σj...σl.
Θέλουμε ένα βήμα για να διαβάσουμε το |> και |x| βήματα για να συμπιέσουμε την είσοδο. Τέλος, θέλουμε και ένα βήμα για να διαβάσουμε το |_| και να καταλάβουμε ότι έχει τελειώσει η είσοδος. Θέλουμε συνολικά **|x| + 2** βήματα.

Στη μηχανή M’ έχουμε k’ ταινίες να περιέχουν ένα |>, εκτός από την πρώτη που περιέχει την είσοδο και τη δεύτερη που έχει τη συμπιεσμένη είσοδο. Θεωρούμε για είσοδο την δεύτερη ταινία και στο τέλος της πρώτης γράφουμε ένα |>, ώστε να τη θεωρήσουμε σαν ταινία εργασίας, ανταλλάσσοντας έτσι ρόλους η πρώτη και η δεύτερη ταινία. Για αυτό το λόγο και όταν k = 1 έχουμε k’ = 2, λόγω της αναγκαστικής ύπαρξης της δεύτερης ταινίας που περιέχει τη συμπιεσμένη είσοδο.
Θα έχουμε κάτι τέτοιο:
$(k_{|\_|}$, |_|, ...) -> (k’, |>, -, ...)
Θέλουμε **ένα** βήμα απλά για να γράψουμε το |> στο τέλος της πρώτης ταινίας (είμαστε ήδη στο τέλος της - $k_{|\_|}$ - μετά το τέλος της συμπίεσης).

Επίσης πρέπει να γυρίσουμε την νέα ταινία είσοδο (τη δεύτερη) στην αρχή της.
Θα έχουμε κάτι τέτοιο:
(k’, |>, σ, ...) -> (k’, |>, -, σ, <-, ...)
(k’, |>, |>, …) -> (s, |>, -, |>, -, …)
με σ ε Σ’ – {|>}

Θέλουμε $\left\lceil \frac{|x|}{m} \right\rceil$ **+ 1** βήματα για να γυρίσουμε πίσω.

Κάθε φορά που η M θα έκανε m πράξεις θα πρέπει η M’ να κάνει λιγότερες πράξεις για να έχουμε καλύτερο χρόνο εκτέλεσης της M’. Θα δείξουμε ότι η M με 6 το πολύ πράξεις προσομοιώνει τις m πράξεις της M. Όπως θα πούμε και πιο μετά, φαίνεται ότι πρέπει να ισχύει m > 6, γιατί αλλιώς κινδυνεύουμε να φτιάξουμε μια μηχανή πιο αργή από αυτή που είχαμε αρχικά. Γιατί 6; Είναι ο αριθμός του έξω από εδώ; Όχι. m κινήσεις της M τις καλούμε «στάδιο». Στην αρχή κάθε σταδίου η μηχανή M’ κινείται μία φορά αριστερά, δύο δεξιά και μετά άλλη μία αριστερά και επιστρέφει στη θέση που ξεκίνησε. Μήπως ζαλίστηκε και κάνει 8άρια; Όλη αυτή την βόλτα την κάνει για να θυμάται τα σύμβολα που είναι αριστερά δεξιά και πάνω στον κέρσορα της κάθε ταινίας. Θυμάται στην πραγματικότητα 3m σύμβολα της M για κάθε ταινία. Με αυτόν τον τρόπο μπορεί και προβλέπει τις κινήσεις της M. Για αυτήν την πρόβλεψη θέλει 4 κινήσεις. Άλλες δύο πράξεις θα χρειαστεί για να εκτελέσει ένα στάδιο, αφού η M το πολύ να μετακινηθεί m θέσεις από εκεί που θα είναι. Άρα, η M’ το πολύ να πάει ένα σύμβολο αριστερά ή δεξιά, δηλαδή να κάνει κάτι εκεί που είναι και κάπου δίπλα, με δύο πράξεις. Με 4 + 2 = 6 πράξεις η Μ΄προσομοιώνει m της Μ.

Πώς όμως η Μ' ξέρει τι πρέπει να κάνει σε ένα στάδιο; Και πώς ξέρει που είναι η Μ σε εκείνο το στάδιο ώστε να την προσομοιώσει. Έχουμε φτιάξει έτσι το πρόγραμμα της Μ', ώστε οι μεταβάσεις δ να αντιστοιχούν σε εκτελέσεις ανά στάδιο. Για κάθε κατάσταση της Μ φτιάχνουμε νέες καταστάσεις $K \times \{1, 2, \ldots, m\}^k \times \Sigma^{3mk}$.
Το Κ αντιστοιχεί στις παλιές καταστάσεις της Μ.
Το $\{1, 2, \ldots, m\}^k$ είναι μία k – άδα που αντιστοιχεί στη θέση του κέρσορα της Μ στο σύμβολο που διαβάζει ο κέρσορας της Μ'. Για παράδειγμα εάν για την i – οστή ταινία ο κέρσορας της Μ βρίσκεται l θέσεις μακριά από το |>, στην i – οστή θέση της k – άδας θα υπάρχει ο αριθμός l mod m.
Το Σ^{3mk} αντιστοιχεί στη σειρά από σύμβολα που θυμάται με τις 4 κινήσεις η Μ', 3m για κάθε μία από τις k ταινίες.
Θα έχουμε κάτι τέτοιο:
((q, …), the_k_string_at_q) -> ((q_l, …), move_cursor_strings_left)
((q_l, …, the_k_string), the_k_string_at_q_l) ->
((q_{lr}, …, the_k_string), move_cursor_strings_right)
((q_{lr}, …, the_k_string), the_k_string_at_q_{lr}) ->
((q_{lrr}, …, the_2*k_string), move_cursor_strings_right)
((q_{lrr}, …, the_2*k_string), the_k_string_at_q_{lrr}) ->
((q_{lrrl}, …, the_3*k_string), move_cursor_strings_left)
με q η κατάσταση στην αρχή κάθε σταδίου, q_l η κατάσταση που θα προκύπτει αν πάνε όλες οι καταστάσεις αριστερά και τα λοιπά.
Οπότε συνολικά έχουμε $|x| + 2 + 1 + \left\lceil \frac{|x|}{m} \right\rceil + 1 + 6\left\lceil \frac{f(|x|)}{m} \right\rceil$ στη χειρότερη περίπτωση.

Αν πάρουμε $m = \left\lceil \frac{6}{\varepsilon} \right\rceil$, έχουμε TIME(|x| + ε 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 χρόνο f(n), τότε μπορούμε να φτιάξουμε άλλη μία τέτοια που να απαιτεί χρόνο εf(n) + n + 2. Δηλαδή εάν η μηχανή Turing θέλει $10n^2$ χρόνο, όπου n είναι το μήκος της εισόδου, τότε μπορούμε να κατασκευάσουμε άλλη μια Turing μηχανή τέτοια που να τρέχει σε $\varepsilon 10n^2 + n + 2$. Για ε = 100 > 0, παίρνουμε ότι η μηχανή μας θα τρέξει σε $\frac{n^2}{10} + n + 2$.

Πρέπει f(n) ≥ n, γιατί αλλιώς θα έχουμε linear slowdown. Εάν για παράδειγμα ο χρόνος εκτέλεσης είναι 100logn για ε = 100 θα γίνει logn + n + 2, χειρότερος.

Παράδειγμα για το LINEAR SPEEDUP theorem.

Είμαι σίγουρος ότι η απόδειξη του SPEEDUP theorem δε σας έπεισε για την ύπαρξη τέτοιας μηχανής. Βασικά, ούτε εμένα με έπεισε. Μπορεί να αποδεικνύεται ότι η μηχανή Μ' που κατασκευάσαμε τρέχει σε χρόνο εf(n) + n + 2, αλλά πουθενά στην απόδειξη δε φαίνεται όλο το πρόγραμμα της Μ'. Δε φαίνεται πουθενά το πρόγραμμα που αφορά τις δύο τελευταίες κινήσεις κάθε σταδίου. Για αυτό θα δώσω ένα παράδειγμα. Έστω ότι η μηχανή Turing με δύο ταινίες που αποφασίζει το εάν μια συμβολοσειρά είναι παλίνδρομη. Έχουμε

Σ = {0, 1, |>, |_|}
K = {s, q, p}
k = 2
m = 7
x = 0110111010111101011110110

P ε K	σ_1 ε Σ	σ_2 ε Σ	Δ(p, σ_1, D_1, σ_2, D_2)
S	0	|_|	(s, 0, ->, 0, ->)
S	1	|_|	(s, 1, ->, 1, ->)
S	|>	|>	(s, |>, ->, |>, ->)
S	|_|	|_|	(q, |_|, <-, |_|, -)
Q	0	|_|	(q, 0, <-, |_|, -)
Q	1	|_|	(q, 1|, <-, |_|, -)
Q	|>	|_|	(p, |>, ->, |_|, <-)
P	0	0	(p, 0, ->, |_|, <-)
P	1	1	(p, 1, ->, |_|, <-)
P	0	1	(“no”, 0, -, 1, -)
P	1	0	(“no”, 1, -, 0, -)
P	|_|	|>	(“yes”, |_|, -, |>, ->)

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

|>																								

Θα πάρουμε
Σ’ = Σ U Σ^7 = Σ U {$\sigma_{0000000}$, $\sigma_{0000001}$, $\sigma_{000000|>}$, $\sigma_{000000|\_|}$, ...} νέα σύμβολα.
1. Μετατρέπουμε την είσοδο σε νέα συμπιεσμένη είσοδο στη ταινία 2.
(k, |>, |>) -> ($k_{|>}$, 0, ->, |>, ->)
($k_{|>}$, 1, |_|) -> ($k_{|>1}$, 0, ->, |_|, -)
($k_{|>1}$, 1, |_|) -> ($k_{|>11}$, 1, ->, |_|, -)
($k_{|>11}$, 0, |_|) -> ($k_{|>110}$, 0, ->, |_|, -)
($k_{|>110}$, 1, |_|) -> ($k_{|>1101}$, 1, ->, |_|, -)
($k_{|>1101}$, 1, |_|) -> ($k_{|>11011}$, 1, ->, |_|, -)
($k_{|>11011}$, 1, |_|) -> ($k_{|>110111}$, 1, ->, |_|, -)
($k_{|>110111}$, 0, |_|) -> (k_0, 0, ->, $\sigma_{|>110111}$, ->), μόλις γράψαμε τα 7 πρώτα σύμβολα σε ένα νέο συμπιεσμένο. Ομοίως για την υπόλοιπη ταινία και για κάθε δυνατή 7 - άδα συμβόλων.

<table>
<tr><td>|></td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>↑</td></tr>
</table>

<table>
<tr><td>|></td><td>$\sigma_{|>110111}$</td><td>$\sigma_{0101111}$</td><td>$\sigma_{0101110}$</td><td>$\sigma_{110|\_||\_||\_||\_|}$</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>↑</td></tr>
</table>

2. Γράφουμε στο τέλος της πρώτης ταινίας το σύμβολο |>.
($k_{|\_|}$, |_| , |_|) -> (k’, |>, -, |_|, -)

<table>
<tr><td>|></td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>|></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>↑</td><td></td></tr>
</table>

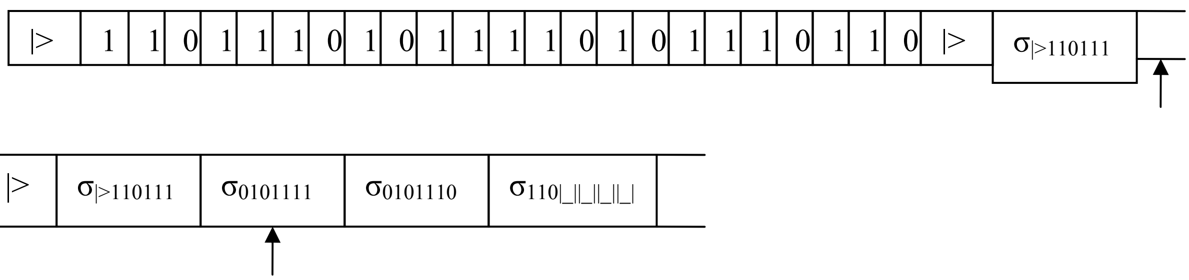
<table>
<tr><td>|></td><td>$\sigma_{|>110111}$</td><td>$\sigma_{0101111}$</td><td>$\sigma_{0101110}$</td><td>$\sigma_{110|\_||\_||\_||\_|}$</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>↑</td></tr>
</table>

3. Γυρνάμε τον κέρσορα της δεύτερης ταινίας ο οποίος βρίσκεται στο τέλος, στην αρχή, στο |>.
(k’, |>, $\sigma_{110|\_||\_||\_||\_|}$) -> (k’, |>, -, $\sigma_{110|\_||\_||\_||\_|}$, <-)
(k’, |>, $\sigma_{0101110}$) -> (k’, |>, -, $\sigma_{0101110}$, <-)
(k’, |>, $\sigma_{0101111}$) -> (k’, |>, -, $\sigma_{0101111}$, <-)
(k’, |>, $\sigma_{|>110111}$) -> (k’, |>, -, $\sigma_{|>110111}$, <-)
(k’, |>, |>) -> (s, (0, 0), |>, -, |>, -), παρόμοιες μεταβάσεις πρέπει να υπάρχουν για όλα τα
7 – σύμβολα.

<table>
<tr><td>|></td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>|></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>↑</td><td></td></tr>
</table>

<table>
<tr><td>|></td><td>$\sigma_{|>110111}$</td><td>$\sigma_{0101111}$</td><td>$\sigma_{0101110}$</td><td>$\sigma_{110|\_||\_||\_||\_|}$</td><td></td></tr>
<tr><td>↑</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

4. Είμαστε στην αρχή σε κάθε ταινία στο |>, πάμε μία θέση δεξιά στη πρώτη θέση κάθε ταινίας, οπότε θα ονοματίσουμε την κατάσταση με τη θέση του κέρσορα στο 7 – μελές σύμβολό μας.
(s, (0, 0), |>, |>) -> ((s, (0, 0)), |>, ->, |>, ->)
5. Τώρα θα πάμε αριστερά,
((s, (0, 0)), |_|, $\sigma_{|>110111}$) -> ((s_l, (0, 0)), |_|, -, $\sigma_{|>110111}$, <-)
δεξιά,
((s_l, (0, 0)), |>, |>) -> ((s_{lr}, (0, 0), (|>,|>)), |>, ->, |>, ->)
πάλι δεξιά,
((s_{lr}, (0, 0), (|>,|>)), |_|, $\sigma_{|>110111}$) -> ((s_{lrr}, (0, 0), (|>|_|,|>$\sigma_{|>110111}$)), |_|, ->, $\sigma_{|>110111}$, ->)
πάρτο αλλιώς – αριστερά,
((s_{lrr}, (0, 0), (|>|_|,|>$\sigma_{|>110111}$)), |_|, $\sigma_{0101111}$) ->
((s_{lrrl}, (0, 0), (|>|_||_|,|>$\sigma_{|>110111}\sigma_{0101111}$)), |_|, <-, $\sigma_{0101111}$, <-)
,παρόμοιες τέτοιες τετράδες μεταβάσεων πρέπει να γράψουμε για κάθε πιθανή κατάσταση.
6. Τώρα ξέρουμε τι υπάρχει αριστερά και δεξιά από τους κέρσορες στις δύο ταινίες μας. Σε πολύ σε δύο βήματα της Μ' προσομοιώνουμε 7 τις Μ.
((s_{lrrl}, (0, 0), (|>|_||_|,|>$\sigma_{|>110111}\sigma_{0101111}$)), |_|, $\sigma_{|>110111}$) ->*
((s, (6, 0)), $\sigma_{110111|\_|}$, ->, $\sigma_{|>110111}$, ->)
,το 6 βήμα είναι και το πιο ζόρικο. Πρέπει για κάθε κατάσταση {s, p, q} και για κάθε δυνατό συνδυασμό των θέσεων των κέρσορων στις ταινίες και για κάθε δυνατό συνδυασμό τριών 7-συμβόλων να γράψουμε μία ή δύο μεταβάσεις της Μ', που θα προσομοιώνουν τις 7 τις Μ. Θέλουμε δηλαδή, $3(7\cdot 7)\binom{4^7}{3}$ (1 or 2) μεταβάσεις, εάν θυμάμαι καλά από συνδυαστική. Θέλουμε 107.732.410.318.848 or 215.464.820.637.696 μεταβάσεις! (Και όταν ξεκίνησα να γράφω αυτό το φυλλάδιο έλεγα να γράψω όλο το πρόγραμμα, α καλό ε;).



...

Υ.Γ.

Κάτι αρκετά ενδιαφέρον θα ήταν το παρακάτω paper, αλλά δε το βρήκα πουθενά, απλά ξέρω ότι υπάρχει.
Viliam Geffert – A speed-up theorem without tape compression. Theoret. Comput. Sci., 118, 49-65, 1993. (Conference version: Proc. Math. Found. Comput. Sci., vol. 452 of Lect. Notes Comput. Sci., pp.285-91. Springer-Verlag, 1990).

Abstract: We shall show that, for each nondeterministic single-tape Turing machine M of time complexity $T(n) \geq n^2$ and each $K \geq 1$ there exists an equivalent K times faster nondeterministic Turing machine M' writing only zeroes and ones on its tape. In other words, we can obtain the linear speed-up while preserving the binary worktape alphabet. Therefore, nondeterministic single-tape Turing machines do not require tape compression for speeding up.

FINE