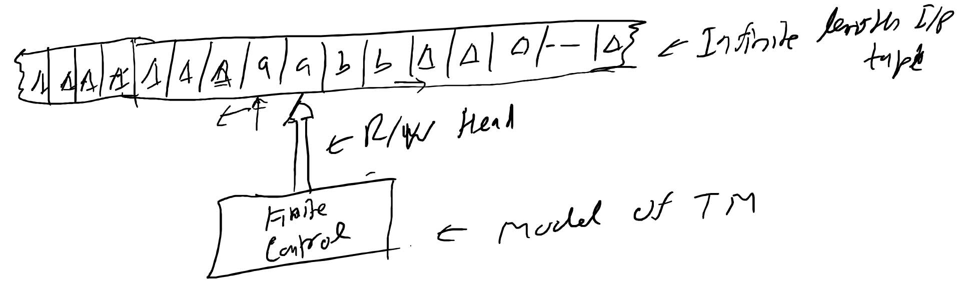
Friday, 8 January, 2021 11:49 AM

Module-4

Turing machines (TOC)

Three address codes (Compiler)

Turing M/C



$M = (Q, \Sigma, \Gamma, \delta, q_0, \Delta, F)$

Q - Non-empty finite set of states

Σ - " " " " " input symbol

Γ - " " " " " tape symbols

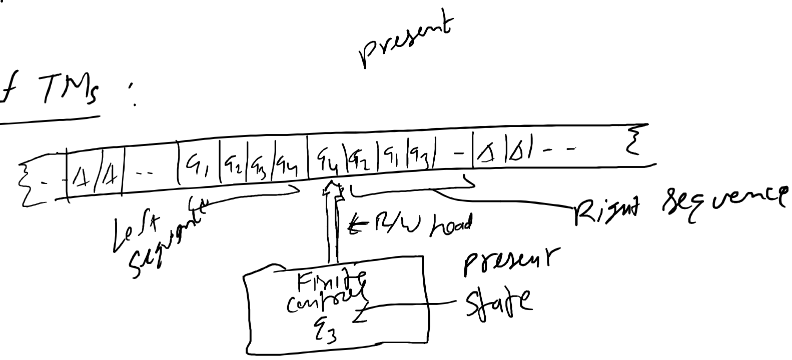
$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times D$ $\quad D = \{R, L\}$

$q_0 \in Q$ - initial state

$\Delta \in \Gamma$ - Blank symbol $(\Delta \notin \Sigma)$

$F \subseteq Q$ - non empty finite set of final states

Representation of TMs :



SNAPSHOT of TM

$w = \underbrace{x_1 x_2 x_3 - x_{i-1}}x_i \underbrace{x_{i+1} - - x_n}, \ |w| = n$

$\delta(q, x_i) = (p, y, R)$ ✓

$(x_1 x_2 - - x_{i-1}\ q\ x_i\ x_{i+1} - - x_n) \vdash (\underbrace{x_1 x_2 - - x_{i-1}\ y}_{\text{Left seq}}\ p\ x_{i+1}\ \underbrace{x_{i+2} - - x_n}_{\text{Right seq}})$ (Right movement)

cur state   cur symb under R/W head

$\delta(q, x_i) = (p, y, L)$ ✓

$(x_1 x_2 - - x_{i-1}\, q\, x_i\, x_{i+1} \ldots)$ — cur state, cur symb under R/W head

$\delta(q, x_i) = (p, y, L)$ ✓

$(x_1 x_2 - - x_{i-1}\, q\, x_i\, x_{i+1} - - x_n) \vdash (x_1 x_2 - - x_{i-2}\, p\, x_{i-1}\, y\, x_{i+1} - - x_n)$ (Left movement)

current ID — next ID

2) Representation by Transition Table

3) Representation by Transition Diagram

Q. Design a TM to accept $L = \{a^n b^n \mid n \geq 1\}$
$= \{ab, aabb, aaabbb, aaaabbbb, \ldots\}$

aaabbb

present state	Tape symbols: a	b	X	Y	Δ
→ q_0	XRq_1 ✓	–	–	YRq_3	
q_1	aRq_1	YLq_2 ✓	–	YRq_1	
q_2	aLq_2	–	XRq_0	YLq_2	
q_3	–	–	–	YRq_3	ΔRq_4
q_4	–	–	–	–	–

aaabbb ⊢ Xaabbb ⊢ XaaYbb ⊢ XaaYbb ⊢ XXaYYb ⊢

XXaYYb ⊢ XXXYYY ⊢ XXXYYY

$q_0$aabb ⊢ X$q_1$abb ⊢ Xa$q_1$bb ⊢ Xq_2aYb ⊢ q_2XaYb ⊢

Xq_0aYb ⊢ XXq_1Yb ⊢ XXYq_1b ⊢ XXq_2YY ⊢ Xq_2XYY ⊢

XXq_0YY ⊢ XXYq_3Y ⊢ XXYYq_3Δ ⊢

XXYYΔq_4Δ — Final State

(halt) ⇒ aabb accepted

abb

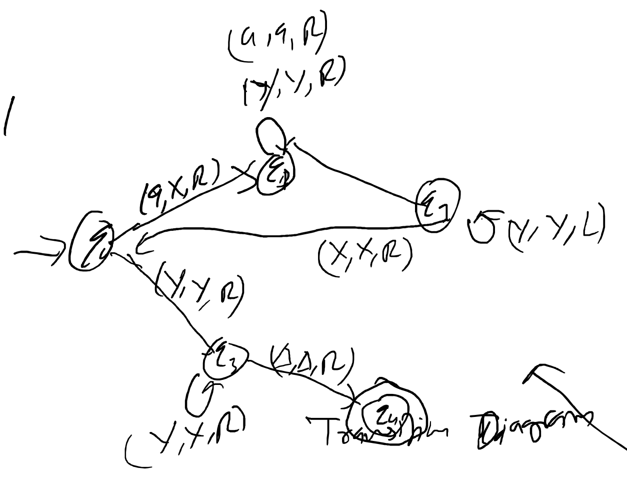
$q_0$abb ⊢ X$q_1$bb ⊢ q_2XYb ⊢ Xq_0Yb ⊢ XYq_3b

M/c halts in non final state by TM

abb does not accepted by TM

by TM

abb does not accepted by TM



Transition Diagram

Transition Table

$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{a,b\}, \{a,b,X,Y,\Delta\}, \delta, q_0, \Delta, \{q_4\})$

$Q, \quad \Sigma, \quad \Gamma$

$\Sigma = \{1\}$, Even no of 1's

$q = \{11, 1111, 111111, \ldots\}$

State	Tape symbol 1	X	Δ
$\rightarrow q_0$	XRq_1	-	-
q_1	XRq_0	-	-

$\underset{q_0}{1111} \vdash \underset{q_1}{XXXX}\Delta$

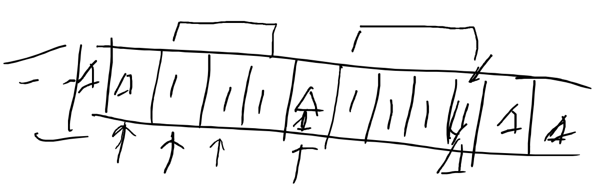
$q_0 1111 \vdash Xq_1 111 \vdash XXq_0 11 \vdash XXXq_1 1 \vdash XXXXq_0\Delta$

$q_0 111 \Rightarrow Xq_1 11 \vdash XXq_0 1 \vdash XXXq_1\Delta$

Q: w_1, w_2

$w_1 = 111 \quad w_2 = 1111$

$w_1 w_2 = 1111111$



States	Tape symbols 1	Δ
	$1Rq_0$	$1Rq_1$
	$1Rq_1$	ΔLq_2

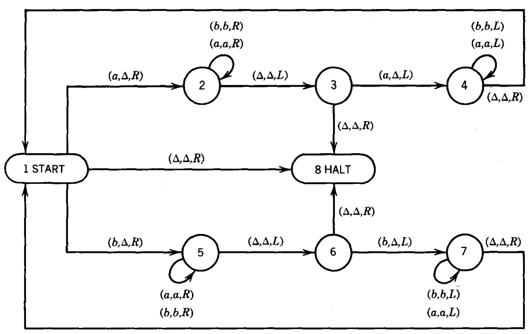
$\delta(q_0, 1) = \{(q_0, 1, R)\}$

	$1Rq_0$	$1Rq_1$
	$1Rq_1$	ΔLq_2
	ΔLq_3	–
q_3	$1Lq_3$	ΔRq_4
q_4	–	–

$\delta(q_0, 1) = \{(q_0, 1, R), (q_1, x, L)\}$

non deterministic TM

TM for palindromes strings over {a, b}



Palindromes

w is a palindrome iff

$w^R = w$

$w = bab \quad w^R = w$

$w^R = bab$

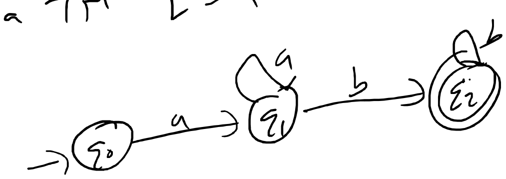
even length (ε, aa, bb, aaaa, abba, baab, bbbb, – –)

odd length (a, b, aaa, aba, bab, bbb, – –)

ababa ✓

$q_0$ababa ⊢ Δ$q_1$baba ⊢ Δ babaq_1Δ ⊢

Q: Design a TM $L = \{a^m b^n \mid m, n \geq 1\}$



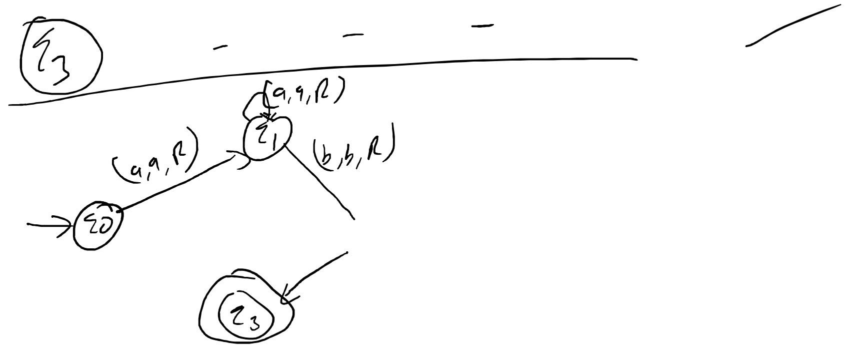
$a^n b^n$

States	a	b	Δ
	aRq_1	–	–
	aRq_1	bRq_2	–
	–	bRq_2	ΔRq_3
q_3	–	–	–

Tape Symbols

aabbb

aabbba

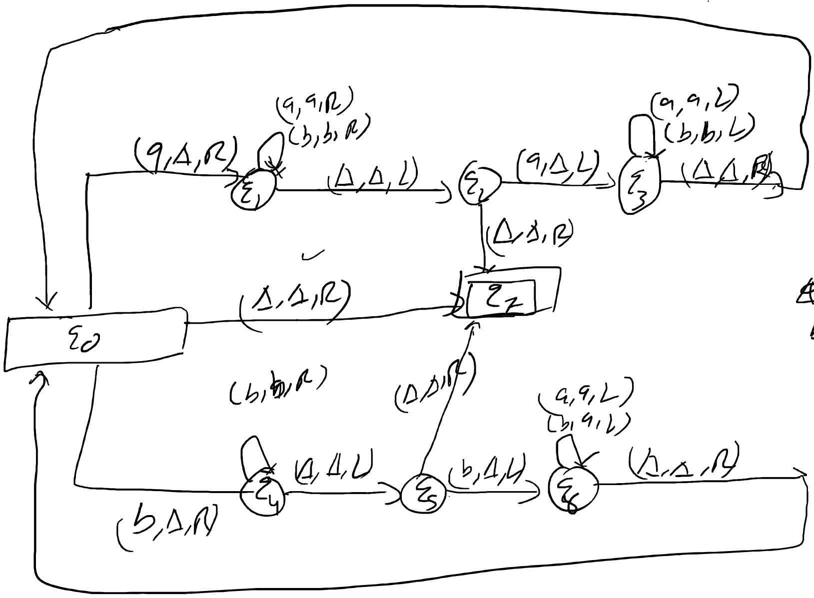


Q: Design a TM to accept all palindromes over $\{a, b\}$

Sol^n: A string w is a palindrome if $w^R = w$

even length palindromes $= \{\epsilon, aa, bb, aaaa, abba, baab, bbbb, \ldots\}$

odd " " $= \{a, b, aaa, aba, bab, bbb, \ldots\}$



babab

baaab

aba

abba

$q_0 aba \vdash \Delta q_1 ba \vdash \Delta b q_1 a \vdash \Delta ba q_1 \Delta \vdash \Delta b q_2 a \Delta \vdash \Delta q_3 b \Delta\Delta \vdash$

$q_3 \Delta b \Delta\Delta \vdash \Delta q_0 b \Delta\Delta \vdash \Delta\Delta q_4 \Delta\Delta \vdash \Delta q_5 \Delta\Delta\Delta \vdash$

$\Delta\Delta q_7 \Delta\Delta$

$q_0 abba \vdash \Delta q_1 bba \vdash \Delta b q_1 ba \vdash \Delta bb q_1 a \vdash \Delta bba q_1 \Delta \vdash$

$\Delta bb q_2 a \Delta \vdash \Delta b q_3 b \Delta\Delta \vdash \Delta q_3 bb \Delta\Delta \vdash$

$q_3 \Delta bb \Delta\Delta \vdash \Delta q_0 bb \Delta\Delta \vdash \Delta\Delta q_4 b \Delta\Delta$

$\Delta\Delta b q_4 \Delta\Delta \vdash \Delta\Delta q_5 b \Delta\Delta \vdash$

$\ldots \Delta\Delta\Delta$

$\varepsilon_3 \Delta bb\Delta\Delta \vdash \Delta\varepsilon_0 bb\Delta\Delta \vdash \Delta\Delta\varepsilon_4 b\Delta\Delta$

$\Delta\Delta b\varepsilon_4\Delta\Delta \vdash \Delta\Delta\varepsilon_5\Delta\Delta\Delta \vdash$

$\Delta\varepsilon_6\Delta\Delta\Delta\Delta \vdash \Delta\Delta\varepsilon_0\Delta\Delta\Delta$

$\Delta\Delta\Delta\varepsilon_7\Delta\Delta$

Three-address codes (Intermediate codes) ✓

$a * -(b+c)$

(i) Postfix notation

(ii) Three address code

$a + b$ (Infix) ✓

$+ab$ (Prefix) ✓

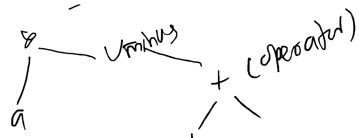
$ab+$ (Postfix) ✓

(i) Postfix notation

Steps $a * -(bc+)$ ✓

$a * (bc+-)$

$a-b$ (Binary) $-a$ (Unary)



Syntax tree

(ii) Three-address code

$a * -(b+c)$

✓ $T_1 := b+c$ ✓ (assignment)

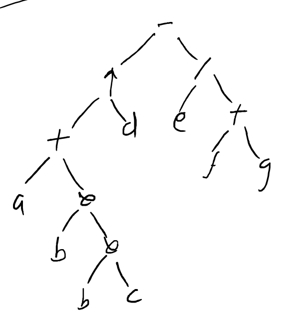
✓ $T_2 := -T_1$ ✓ (")

✓ $T_3 := a * T_2$ ✓ (")

$x := y$ (copy)

Q. Construct syntax tree and postfix notation for the expression

$x = \{[a + b * (b * c)] \uparrow d - \{e/(f+g)\}\}$



(Syntax tree)

$2^3 = 2 \uparrow 3$

(Syntax tree)

$(a + b*(b*c)) \uparrow d - e/(f+g)$

$(a + b*(b*c)) \uparrow d - e/(f+g)$

$T_1 := b*c$ ✓

$T_2 := b*T_1$ ✓ assignment

$T_3 := a+T_2$

$(a+b*(b*c))\uparrow d - e/(f+g)$

$(a+b*(bc*))\uparrow d - e/(f+g)$

Let $T_1 = bc*$

$(a+b*T_1)\uparrow d - e/(f+g)$

$(a+(bT_1*))\uparrow d - e/(f+g)$

Let $T_2 = bT_1*$

$(a+T_2)\uparrow d - e/(f+g)$

$(aT_2+)\uparrow d - e/(fg+)$

Let $T_3 = aT_2+$, $T_4 = fg+$

$T_3\uparrow d - e/T_4$

$(T_3d\uparrow) - (eT_4/)$

$T_3d\uparrow eT_4/-$

$aT_2+d\uparrow e\,fg+-$

$abT_1*+d\uparrow e\,fg+-$

$\boxed{abbc**+d\uparrow e\,fg+-}$

$T_1 := b$

$T_2 := b*T_1$ ✓ assign

$T_3 := a+T_2$ ✓

$T_4 := T_3\uparrow d$ ✓

$T_5 := f+g$

$T_6 := e/T_5$

$T_7 := T_4 - T_6$

3-address code

Q: Translate the expression $-(a+b)*(c+d) + (a+b+c)$ into
(i) Quadruples (ii) Triples (iii) Indirect Triples

$T_1 := a+b$ ✓

$T_2 := -T_1$

$T_3 := c+d$

$T_4 := T_2 * T_3$

$T_5 := T_1 + c$

$T_6 := T_4 + T_5$

	operator	operand 1	operand 2	Result
(1)	+	a	b	T_1
(2)	−	T_1		T_2
(3)	+	c	d	T_3
(4)	*	T_2	T_3	T_4
(5)	+	T_1	c	T_5
(6)	+	T_4	T_5	T_6

Quadruples

(ii) Triples

	operator	operand 1	operand 2

(ii) Triples

	operator	operand 1	operand 2
(1)	+	a	b
(2)	−	(1)	
(3)	+	c	d
(4)	&	(2)	(3)
(5)	+	(1)	c
(6)	+	(4)	(5)

(Triples)

Indirect Triples

	Statement
(100)	(1)
(200)	(2)
(300)	(3)
(400)	(4)
(500)	(5)
(600)	(6)

	operator	op1	op2
(100)	+	a	b
(200)	−	(100)	
(300)	+	c	d
(400)	&	(200)	(300)
(500)	+	(100)	c
(600)	+	(400)	(500)

$\boxed{i=10}$ $\boxed{p=100}$ 200 $\boxed{100}$ r=200

int i = 10;
int *p;
int **r;

p = &i;
r = &p

printf("%d", i) = 10
printf(%d, *p) = 10
printf(%d, **r) = 10

Q: Construct three-address codes for the following:

if [(a<b) and ((c>d) or (a>d))] then
z = x+y+z
else z = x+1

(1) if a<b goto (3)
(2) goto (11)
(3) if c>d goto (7)
(4) goto (5)
(5) if a>d goto (7)
(6) goto (11)
(7) $t_1 := x+y$
(8) $t_2 := t_1+z$
(9) $z := t_2$
(10) goto (13)
(11) $t_3 := x+1$

(10) goto (13)

(11) $t_3 := x+1$

(12) $z := t_3$

(13) Exit

Q: while a<b do
    if c<d then
      x := y+z
    else
      x := y−z

Soln

(1) if a<b goto (3)

(2) goto (11)

(3) if c<d goto (5)

(4) goto (8)

(5) $t_1 := y+z$

(6) $x := t_1$

(7) goto (1)

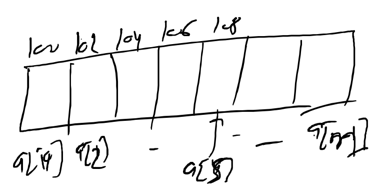
(8) $t_2 := y-z$

(9) $x := t_2$

(10) goto (1)

(11) Exit

Q: main( )
{
  int i=1;
  int a[10];
  while (i<=10)
  a[i] = i;
  i=i+1;



$w = 2$

$a[i] = base(a) + (i-1) \times w$

$a[5] = 100 + 4 \times 2 = 108$

$a[i] = base(a) + (i-1)w$

$= base(a) + i \times w - w$

$a[i] = base(a) - w + i \times w$

$t_1 = i \times w$ (offset)

$t_2 := base(a) - w$ (base address)

$a[i] = t_2[t_1]$

(1) i = 1

(2) if i<=10 goto (4)   x := y1

(3) goto (10)

(4) $t_1 := i \times 4$

Let w = 4 bytes/word

(5) $t_2 := base(a) - 4$

(6) $t_2[t_1] = i$;

(7) $t_3 := i+1$

(8) $i = t_3$

(9) goto (2)

(8) i = t_3

(9) goto (2)

(10) Exit

Q. Write 3-address code for the following program fragment

```
sum: = 0
for ( i = 1 ; i <= 20 ; i = i+1 )
   sum: = sum + a[i] + b[i] .
```

(1) sum = 0

(2) i = 1

(3) if i <= 20 goto (5)

(4) goto (17)

(5) t_1 := i × 4

(6) t_2 := base(a) − 4

(7) t_3 := $t_2[t_1]$

(8) t_4 := i × 4

(9) t_5 := base(b) − 4

(10) t_6 := $t_5[t_4]$

(11) t_7 := sum + t_3

(12) t_8 := $t_7 + t_6$

(13) sum = t_8

(14) t_9 := i + 1

(15) i := t_9

(16) goto (3)

(17) Exit

w = 4 bytes/word

p = &q

q = 100, q = 10

and &q

r = &q;

Q. while (A < C and B > D) do

if A = 1 then C := C + 1

else while A <= D do

A := A + 3

(1) if A<C goto (3)
(2) goto (15)
(3) if B>D goto (5)
(4) goto (15)
(5) if A=1 goto (7)
(6) goto (10)
(7) $t_1 := C+1$
(8) $C := t_1$
(9) goto (1)
(10) if A<=D goto (12)
(11) goto (1)
(12) $t_2 := A+3$
(13) $A := t_2$
(14) goto (10)
(15) Exit

Q//
$x = a[i]+1$
$a[i] = b[c[i]]$
$a[i][j] = b[i][k] \times c[k][j]$
$a[i] = a[i] + b[j]$

$d_1 \times d_2$
10×20
$\uparrow d_1 \quad \uparrow d_2$

(1) $t_1 = i \times 4$
(2) $t_2 = base(a) - 4$
(3) $t_3 := t_2[t_1]$
(4) $t_4 := t_3 + 1$
(5) $x := t_4$
(6) $t_5 := i \times 4$
(7) $t_6 := base(c) - 4$
(8) $t_7 := t_6[t_5] \leftarrow c[i]$
(9) $t_8 := t_7 \times 4$
(10) $t_9 := base(b) - 4$
(11) $t_{10} := t_9[t_8] \leftarrow b[c[i]]$
(12) $t_2[t_1] = t_{10}$

$w = $ 4 bytes/word

$(d_1 \times d_2 = 10 \times 20)$
$(w = 4)$

$a[2][1]$

$t_1 = 2 \times 3 = 6$
$t_2 = 6+1 = 7$
$t_3 = 7 \times 2 = 14$
$t_4 = 100 - 8 = 92$

$$\left[\begin{array}{l} t_1 = i \times d_2 \\ t_2 = t_1 + j \\ t_3 = t_2 \times w \\ t_4 = add(a) - C \\ a[i][j] = t_4[t_3] \end{array}\right]$$

$C = (d_1+1) \times w$
$= (3+1) \times 2$
$= 8$

100	102	104
a[1][1]	a[1][2]	a[1][3]
a[2][1]	a[2][2]	a[2][3]
106	108	110

$d_1 \times d_2 = 2 \times 3 \quad (d_1 = 2, d_2 = 3)$
$w = 2$
$t_1 = i \times d_2 = 2 \times 3 = 6$
$t_2 = t_1 + j = 6 + 1 = 7$

$C = (d_2+1) \times w$
$= (3+1) \times 2$

(12) $t_2[t_1] = t_{10}$

(13) $t_{11} = i \times 20$

(14) $t_{12} = t_{11} + j$

(15) $t_{13} = t_{12} \times 4$

(16) $t_{14} = add(a) - 84$

(17) $t_{15} = i \times 20$

(18) $t_{16} = t_{15} + k$

(21) $t_{19} = t_{18}[t_{17}]$

(22) $t_{20} = k \times 20$

(23) $t_{21} = t_{20} + j$

(24) $t_{22} = t_{21} \times w$

(25) $t_{23} = add(c) - 84$

(26) $t_{24} = t_{23}[t_{22}]$

(27) $t_{25} = t_{19} \times t_{24}$ ✓

(30) $t_{14}[t_{13}] = t_{25}$

(31) $t_{26} = j \times 4$

(32) $t_{27} = add(b) - 4$

(33) $t_{28} = t_{27}[t_{26}]$

(34) $t_{29} = t_2[t_1]$

(35) $t_{30} = t_{28} + t_{29}$

(36) $t_2[t_1] = t_{30}$ ✓

$t_1 = i \times d_2 = 2 \times 3 = 6$

$t_2 = t_1 + j = 6 + 1 = 7$

$t_3 = t_2 \times w = 7 \times 2 = 14$

$t_4 = add(a) - C = 100 - 8 = 92$

$a[i][j] = a[2][1] = t_4[t_3] = 92 + 14 = 106$

$C = (d_2 + 1) \times w$
$= (3+1) \times 2$
$= 8$

$$a[i][j] = t_4[t_3]$$
$$= \underbrace{add(a) - (d_2+1) \times w}_{\leftarrow t_4}$$
$$+ (i \times d_2 + j) \times w \leftarrow t_3$$
$$= add(a) - d_2 \times w - w$$
$$+ i \times d_2 \times w + j \times w$$
$$= add(a) + [-(d_2+1) + (i \times d_2 + j)] \times w$$

$$\boxed{a[i][j] = add(a) + [(i \times d_2 + j) - (d_2 + 1)] \times w}$$

$$a[i][j] = add(a) + (i \times d_2 + j) \times w - C \qquad C = (d_2+1) \times w$$
$$= \underbrace{(add(a) - C)}_{t_4} + \underbrace{(i \times d_2 + j) \times w}_{t_3}$$

$t_1 = i \times d_2$

$t_2 = t_1 + j$

$t_3 = t_2 \times w$

$t_4 = add(a) - C$

$a[i][j] = t_4[t_3]$

$C = (d_2 + 1) \times w$