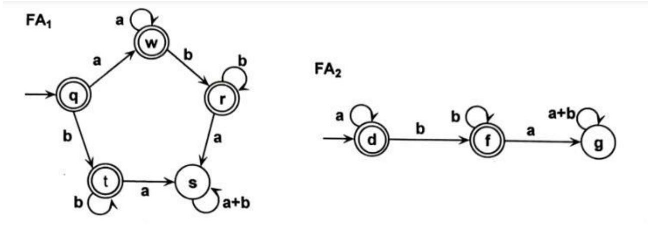
Mark Niño C. Ursal
BS Computer Science – 3rd Year

CS 3143 Automata Theory and Formal Languages
(4-314)

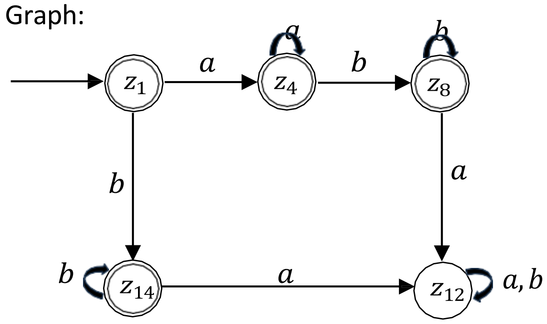
CS281: Introduction to Automata Theory
Intersection Equivalence Seatwork

1.

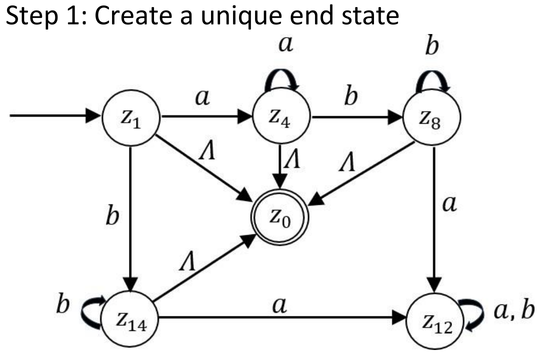


Z	X	Y	FA_1'	FA_2'	$FA_1' + FA_2'$	$(FA_1' + FA_2')'$	X	Y	XY	X	Y	XY
z_1	q	d	-	-	-	±	w	d	z_4	t	f	z_{14}
z_2	q	f	-			+	w	g	z_6	t	f	z_{14}
z_3	q	g	-	+	+		w	g	z_6	t	g	z_{15}
z_4	w	d		-		+	w	d	z_4	r	f	z_8
z_5	w	f				+	w	g	z_6	r	f	z_8
z_6	w	g		+	+		w	g	z_6	r	g	z_9
z_7	r	d		-		+	s	d	z_{10}	r	f	z_8
z_8	r	f				+	s	g	z_{12}	r	f	z_8
z_9	r	g		+	+		s	g	z_{12}	r	g	z_9
z_{10}	s	d	+	-	+		s	d	z_{10}	s	f	z_{11}
z_{11}	s	f	+		+		s	g	z_{12}	s	f	z_{11}
z_{12}	s	g	+	+	+		s	g	z_{12}	s	g	z_{12}
z_{13}	t	d		-		+	s	d	z_{10}	t	f	z_{14}
z_{14}	t	f				+	s	g	z_{12}	t	f	z_{14}
z_{15}	t	g		+	+		s	g	z_{12}	t	g	z_{15}

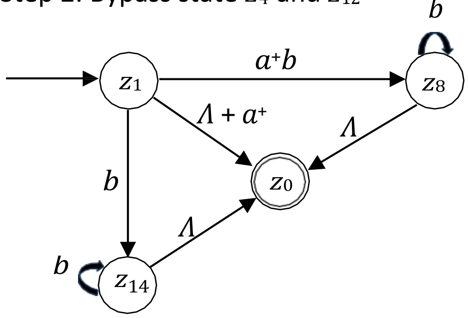
Graph:



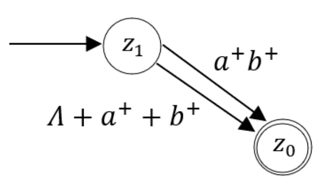
Step 1: Create a unique end state



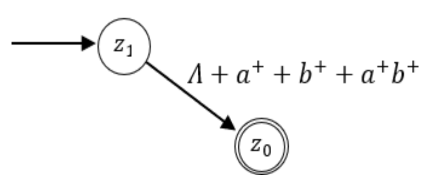
Step 2: Bypass state z_4 and z_{12}



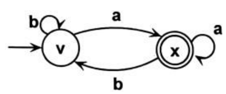
Step 3: Bypass state z_{14} and z_8

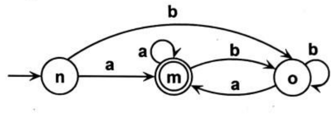


Step 4: Simplify



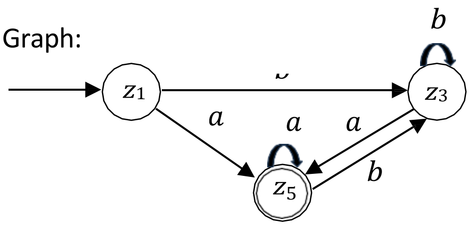
2. FA$_1$ FA$_2$



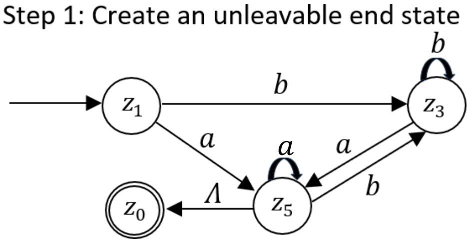


Z	X	Y	FA_1'	FA_2'	$FA_1'+FA_2'$	$(FA_1'+FA_2')'$	X	Y	XY	X	Y	XY
z_1	v	n	±	±	±	-	x	m	z_5	v	o	z_3
z_2	v	m	±		+		x	m	z_5	v	o	z_3
z_3	v	o	±	+	+		x	m	z_5	v	o	z_3
z_4	x	n		±	+		x	m	z_5	v	o	z_3
z_5	x	m				+	x	m	z_5	v	o	z_3
z_6	x	o		+	+		x	m	z_5	v	o	z_3

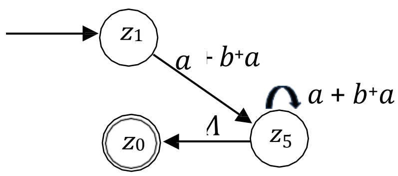
Graph:



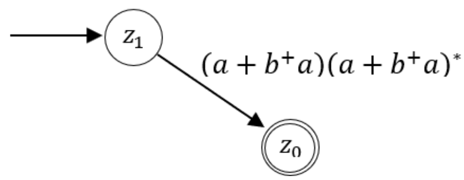
Step 1: Create an unleavable end state



Step 2: Bypass state z_3

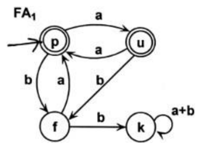


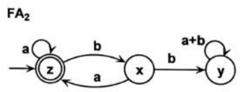
Step 3: Bypass state z_5



Mark Niño C. Ursal
BS Computer Science – 3rd Year

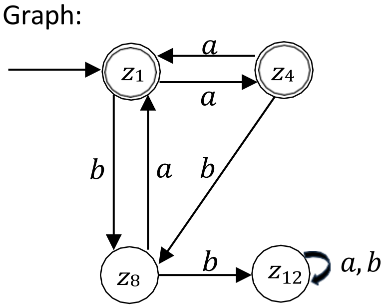
3.



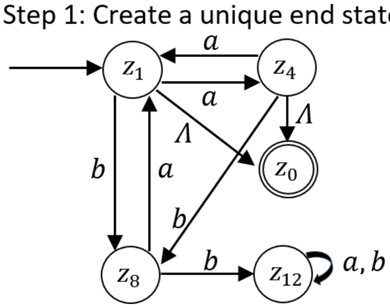


Z	X	Y	FA_1'	FA_2'	$FA_1' + FA_2'$	$(FA_1' + FA_2')'$	X	Y	XY	X	Y	XY
z_1	p	z	-	-	-	±	u	z	z_4	f	x	z_8
z_2	p	x	-	+	+		u	z	z_4	f	y	z_9
z_3	p	y	-	+	+		u	y	z_6	f	y	z_9
z_4	u	z		-		+	p	z	z_1	f	x	z_8
z_5	u	x		+	+		p	z	z_1	f	y	z_9
z_6	u	y		+	+		p	y	z_3	f	y	z_9
z_7	f	z	+	-	+		p	z	z_1	k	x	z_{11}
z_8	f	x	+	+	+		p	z	z_1	k	y	z_{12}
z_9	f	y	+	+	+		p	y	z_3	k	y	z_{12}
z_{10}	k	z	+	-	+		k	z	z_{10}	k	x	z_{11}
z_{11}	k	x	+	+	+		k	z	z_{10}	k	y	z_{12}
z_{12}	k	y	+	+	+		k	y	z_{12}	k	y	z_{12}

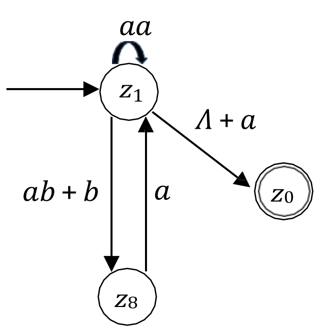
Graph:



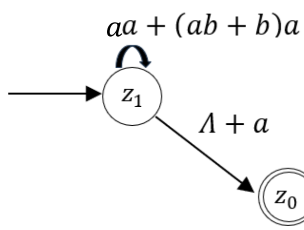
Step 1: Create a unique end state



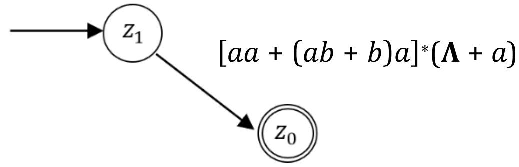
Step 2: Bypass state z_4 and z_{12}



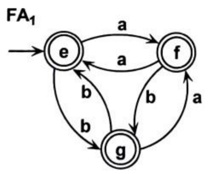
Step 3: Bypass state z_8

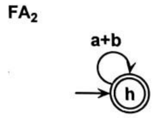


Step 4: Simplify



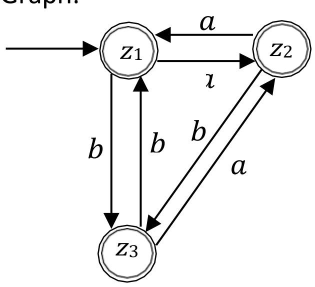
4.



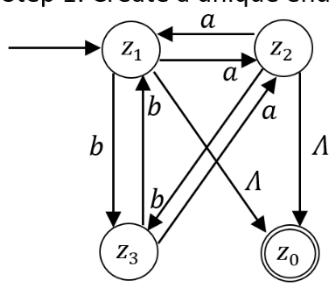


Z	X	Y	FA_1'	FA_2'	$FA_1' + FA_2'$	$(FA_1' + FA_2')'$	X	Y	XY	X	Y	XY
z_1	e	h	-	-	-	±	f	h	z_2	g	h	z_3
z_2	f	h		-		+	e	h	z_1	g	h	z_3
z_3	g	h		-		+	f	h	z_2	e	h	z_1

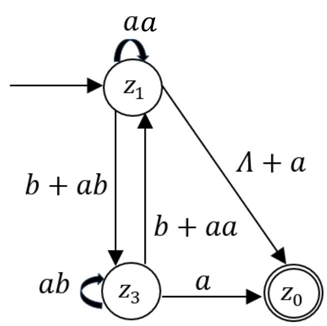
Graph:



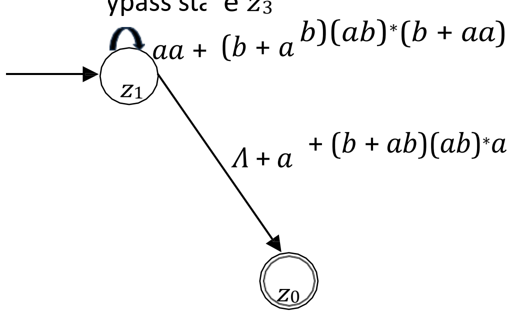
Step 1: Create a unique end state



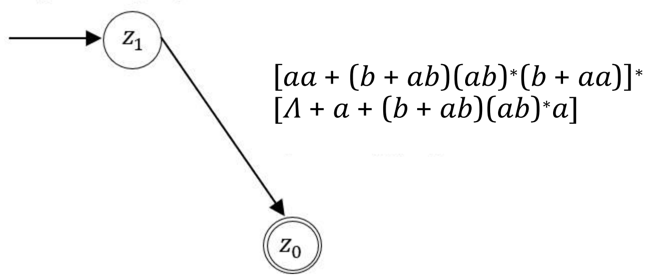
Step 2: Bypass state z_2



Step 3: Bypass state z_3

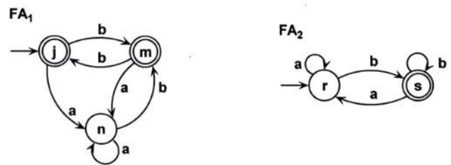


Step 4: Simplify



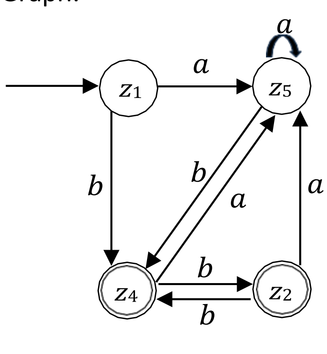
Mark Niño C. Ursal
BS Computer Science – 3rd Year

5.

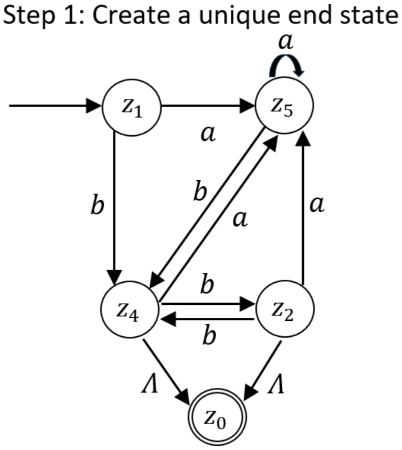


Z	X	Y	FA_1'	FA_2'	$FA_1'+FA_2'$	$(FA_1'+FA_2')'$	X	Y	XY	X	Y	XY
z_1	j	r	-	±	±	-	n	r	z_5	m	s	z_4
z_2	j	s	-			+	n	r	z_5	m	s	z_4
z_3	m	r		±	+		n	r	z_5	j	s	z_2
z_4	m	s				+	n	r	z_5	j	s	z_2
z_5	n	r	+	±	+		n	r	z_5	m	s	z_4
z_6	n	s	+		+		n	r	z_5	m	s	z_4

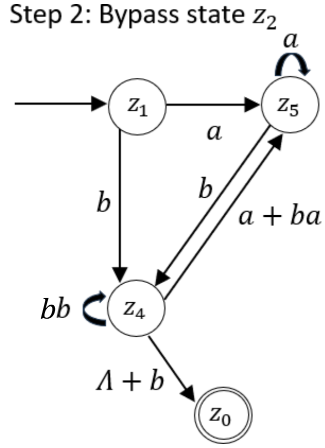
Graph:



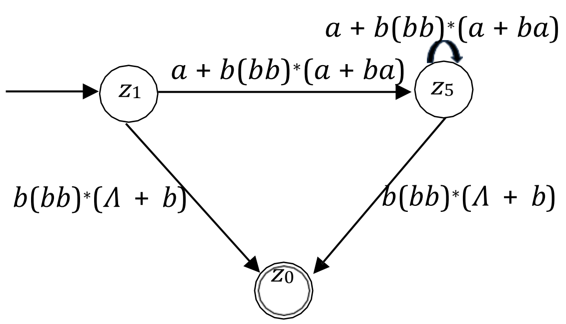
Step 1: Create a unique end state



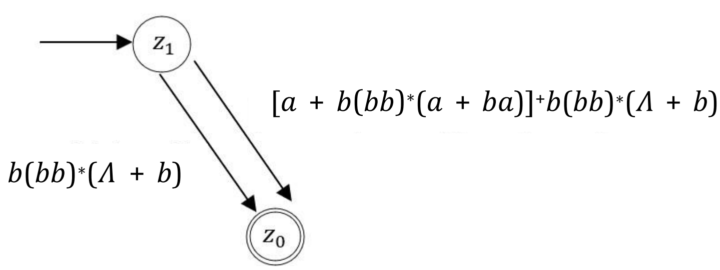
Step 2: Bypass state z_2



Step 3: Bypass state z_4



Step 4: Bypass state z_5



Step 5: Simplify