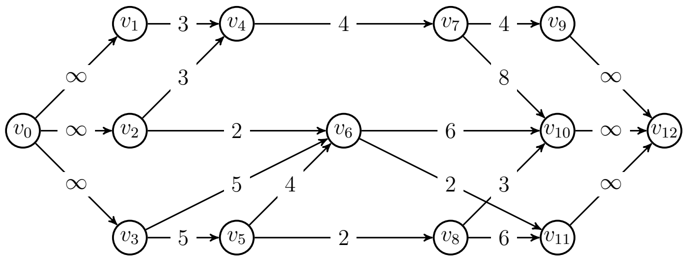
MA4016 - Engineering Mathematics 6

Problem Sheet 11: Maximal Flow and Turing machines

(April 23, 2010)

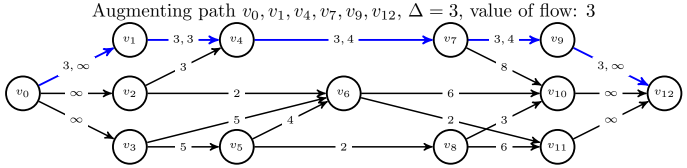
1. Find a maximal flow and a minimal cut in the following pumping network.



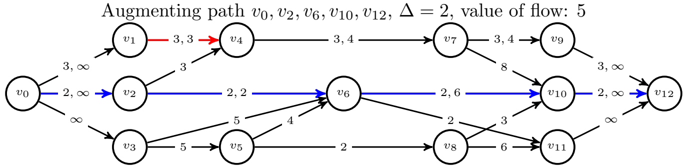
Find a maximal flow after changing the following labels: $(v_6, v_{11}) = 4$, $(v_4, v_7) = 8$.

A minimal cut is $P = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6\}$, $P' = \{v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}$ with $C(P, P') = 14$. We use the Ford-Fulkerson algorithm and increase the flow according to the following pictures. On edges with no flow given assume a flow of 0.

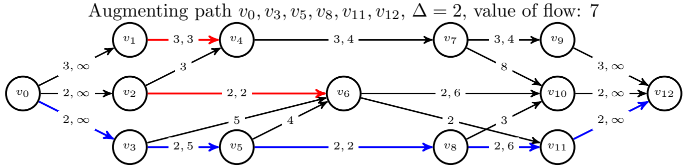
Augmenting path $v_0, v_1, v_4, v_7, v_9, v_{12}$, $\Delta = 3$, value of flow: 3



Augmenting path $v_0, v_2, v_6, v_{10}, v_{12}$, $\Delta = 2$, value of flow: 5



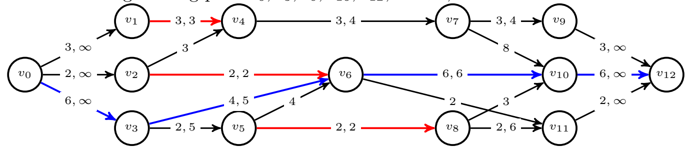
Augmenting path $v_0, v_3, v_5, v_8, v_{11}, v_{12}$, $\Delta = 2$, value of flow: 7



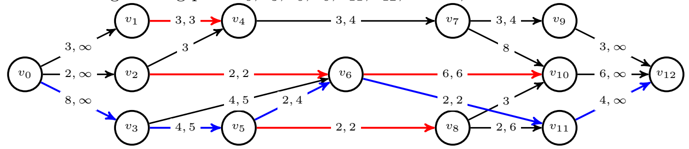
University of Limerick
Department of Mathematics and Statistics
Dr. S. Franz

OLLSCOIL LUIMNIGH
UNIVERSITY OF LIMERICK

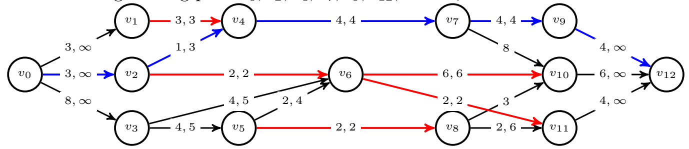
Augmenting path $v_0, v_3, v_6, v_{10}, v_{12}$, $\Delta = 4$, value of flow: 11



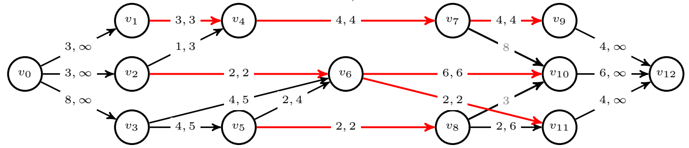
Augmenting path $v_0, v_3, v_5, v_6, v_{11}, v_{12}$, $\Delta = 2$, value of flow: 13



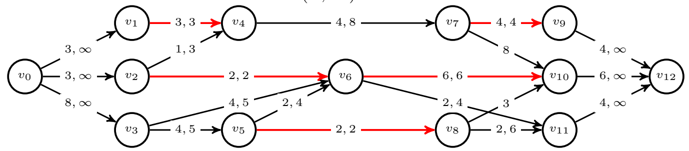
Augmenting path $v_0, v_2, v_4, v_7, v_9, v_{12}$, $\Delta = 1$, value of flow: 14



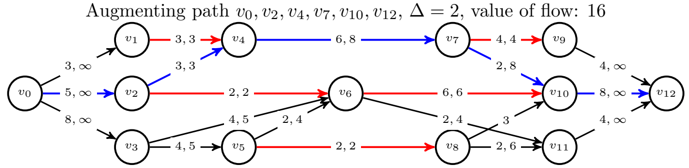
maximal flow, value of 14



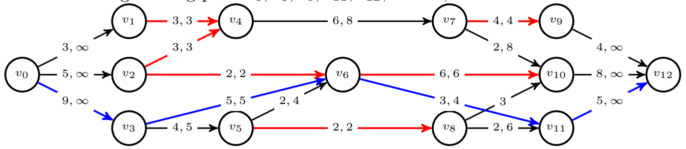
After change of labels the flow in the previous network is still a flow in the new network. We use it as initial flow. A new minimal cut is $P = \{v_0, v_1, v_2, v_3, v_5, v_6\}$, $P' = \{v_4, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}$ with $C(P, P') = 18$.



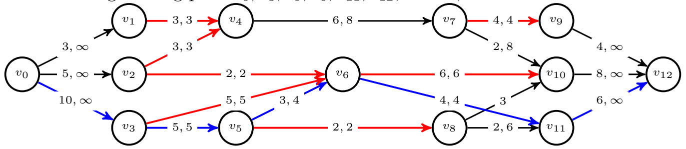
Augmenting path $v_0, v_2, v_4, v_7, v_{10}, v_{12}$, $\Delta = 2$, value of flow: 16



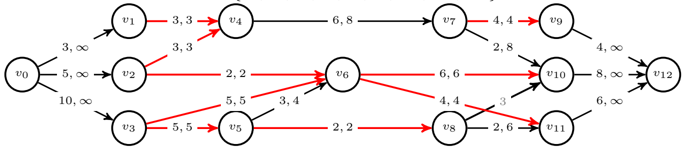
Augmenting path $v_0, v_3, v_6, v_{11}, v_{12}$, $\Delta = 1$, value of flow: 17



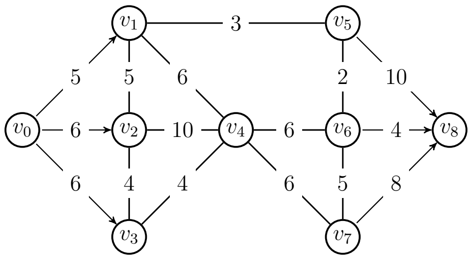
Augmenting path $v_0, v_3, v_5, v_6, v_{11}, v_{12}$, $\Delta = 1$, value of flow: 18



maximal flow with value 18, another minimal cut is $P = \{v_0, v_1, v_2, v_3\}$ and $P' = \{v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}$

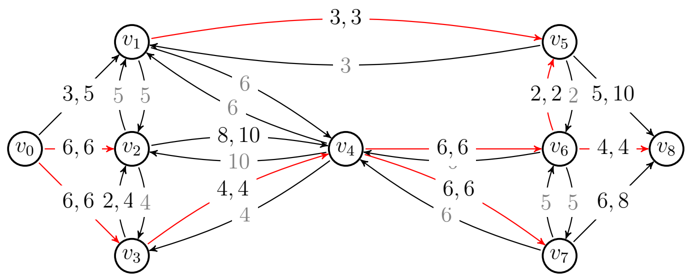


2. We want to maximise the flow from v_0 to v_8. The flow between two vertices, neither of which is v_0 or v_8, can be in either direction. Model this as system as a network and find a maximal flow and a minimal cut.

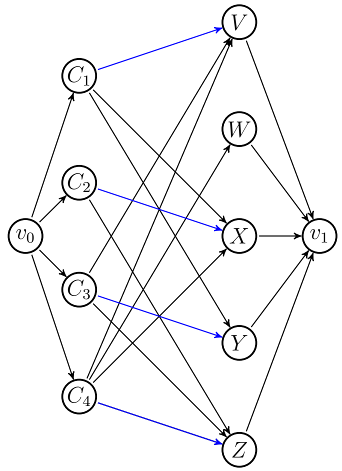


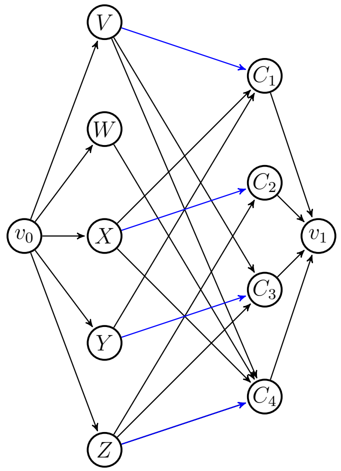
We replace each undirected labelled edge by two directed edges pointing in opposite directions with the same label. In this network graph we use the Ford-Fulkerson algorithm to find a maximal flow. A minimal cut is given by $P = \{v_0, v_1, v_2, v_3, v_4\}$ and $P' = \{v_5, v_6, v_7, v_8\}$ with $C(P, P') = 15$. Applying the algorithm gives the maximum flow in the following picture.

University of Limerick
Department of Mathematics and Statistics
Dr. S. Franz
OLLSCOIL LUIMNIGH
UNIVERSITY OF LIMERICK



3. Five students, V, W, X, Y, and Z, are members of four committees, C_1, C_2, C_3, and C_4. The members of C_1 are V, X, and Y; the members of C_2 are X and Z; the members of C_3 are V, Y, and Z; and the members of C_4 are V, W, X, and Z. Each committee is to send a representative to the administration. No student can represent more than one committee.
Model this situation in a matching network and find a maximal matching. Is this a complete matching?
The corresponding matching network with oriented edges from the committees to the students and a complete matching in blue is given in the figure to the left. If we model the problem with oriented edges from the students to the committees, we find a maximal but not complete matching, see figure to the right. It is the same matching, so "complete" depends on the model.





4. Construct a Turing machine with tape symbols 0, 1, and B, when given a bit string as input, adds a 1 to the end of the bit string and does not change any of the other symbols on the tape.

add 1	0	1	B
s_0	s_0 0 R	s_0 1 R	s_1 1 R

5. Construct a Turing machine with tape symbols 0, 1, and B, when given a bit string as input, replaces the first 0 with a 1 and does not change any of the other symbols on the tape.

first 0 $\to$ 1	0	1	B
s_0	s_1 1 R	s_0 1 R	s_1 B R

6. Construct a Turing machine that recognises the set of all bit strings that end with a 0.

**0	0	1	B
s_0	s_0 0 R	s_0 1 R	s_1 B L
s_1	s_2 0 R		

s_2 is final state, s_1 is non final state. String is recognised as ending with 0 if Turing machine stops in state s_2.

7. Construct a Turing machine that computes the function $f(n) = n + 2$ for all nonnegative integers n.
We represent $n \geq 0$ in the unary system by $n + 1$ 1's on the tape and append two more 1's to the string.

add 2	1	B
s_0	s_0 1 R	s_1 1 R
s_1		s_2 1 R

8. Construct a Modulo-4 Machine, i.e. a *Turing* Machine which takes as tape input a string of symbols representing an integer and produces as tape output the string of symbols representing the remainder after division by 4 of the integer.
Illustrate the operation of the Modulo-4 Machine on the input string representing the number *seven*.

First let us use the unary system—encoding of n with $n+1$ 1's on the tape. We have to check, whether 5=4+1 1's are on the tape. If not, we are finished. If they are on the tape, we have to erase the left four 1's (idea: n mod 4 =$(n-4)$ mod 4).
We get a more compact Turing machine if we use a different encoding of n. Here a suitable one is the representation of n to the basis 4, e.g. $4 = (10)_4$, $7 = (13)_4$ ... Our alphabet is 0, 1, 2, 3, B. We have to check for two non-blank symbols on the tape and erase the left one (idea: n mod 4=$(n-4k)$ mod 4 for integers k).

University of Limerick
Department of Mathematics and Statistics
Dr. S. Franz

OLLSCOIL LUIMNIGH
UNIVERSITY OF LIMERICK

Turing machine using the encoding with $n+1$ 1's for n.

mod 4	1	B	
s_0	s_1 1 R		check for 5 1's
s_1	s_2 1 R		
s_2	s_3 1 R		
s_3	s_4 1 R		
s_4	s_5 1 L		
s_5	s_6 B L		delete four leftmost 1's
s_6	s_7 B L		
s_7	s_8 B L		
s_8	s_9 B R		
s_9		s_{10} B R	go to new start of string
s_{10}		s_{11} B R	and restart Turing machine
s_{11}		s_0 B R	

First Turing machine applied to the encoding $7 = 11111111$

tape	state	rule
...B $\boxed{1}$ 1 1 1 1 1 1 1 B ...	s_0	$(s_0, 1, s_1, 1,\mathrm{R})$
...B 1 $\boxed{1}$ 1 1 1 1 1 1 B ...	s_1	$(s_1, 1, s_2, 1,\mathrm{R})$
...B 1 1 $\boxed{1}$ 1 1 1 1 1 B ...	s_2	$(s_2, 1, s_3, 1,\mathrm{R})$
...B 1 1 1 $\boxed{1}$ 1 1 1 1 B ...	s_3	$(s_3, 1, s_4, 1,\mathrm{R})$
...B 1 1 1 1 $\boxed{1}$ 1 1 1 B ...	s_4	$(s_4, 1, s_5, 1,\mathrm{L})$
...B 1 1 1 $\boxed{1}$ 1 1 1 1 B ...	s_5	$(s_5, 1, s_6,\mathrm{B},\mathrm{L})$
...B 1 1 $\boxed{1}$ B 1 1 1 1 B ...	s_6	$(s_6, 1, s_7,\mathrm{B},\mathrm{L})$
...B 1 $\boxed{1}$ B B 1 1 1 1 B ...	s_7	$(s_7, 1, s_8,\mathrm{B},\mathrm{L})$
...B $\boxed{1}$ B B B 1 1 1 1 B ...	s_8	$(s_8, 1, s_9,\mathrm{B},\mathrm{R})$
...B B $\boxed{B}$ B B 1 1 1 1 B ...	s_9	$(s_9,\mathrm{B}, s_{10},\mathrm{B},\mathrm{R})$
...B B B $\boxed{B}$ B 1 1 1 1 B ...	s_{10}	$(s_{10},\mathrm{B}, s_{11},\mathrm{B},\mathrm{R})$
...B B B B $\boxed{B}$ 1 1 1 1 B ...	s_{11}	$(s_{11},\mathrm{B}, s_0,\mathrm{B},\mathrm{R})$
...B B B B B $\boxed{1}$ 1 1 1 B ...	s_0	$(s_0, 1, s_1, 1,\mathrm{R})$
...B B B B B 1 $\boxed{1}$ 1 1 B ...	s_1	$(s_1, 1, s_2, 1,\mathrm{R})$
...B B B B B 1 1 $\boxed{1}$ 1 B ...	s_2	$(s_2, 1, s_3, 1,\mathrm{R})$
...B B B B B 1 1 1 $\boxed{1}$ B ...	s_3	$(s_3, 1, s_4, 1,\mathrm{R})$
...B B B B B 1 1 1 1 $\boxed{B}$ ...	s_4	no rule $\to$ stop

Turing machine using the encoding of n to the basis 4.

mod 4	0	1	2	3	B	
s_0	s_1 0 R	s_1 1 R	s_1 2 R	s_3 3 R		check for two non-blank symbols
s_1	s_2 0 L	s_2 1 L	s_2 2 L	s_2 3 L		
s_2	s_0 B R	s_0 B R	s_0 B R	s_0 B R		delete left one and restart

Second Turing machine applied to the encoding $7 = (13)_4$

tape	state	rule
...B $\boxed{1}$ 3 B ...	s_0	$(s_0, 1, s_1, 1,\mathrm{R})$
...B 1 $\boxed{3}$ B ...	s_1	$(s_1, 3, s_2, 3,\mathrm{L})$
...B $\boxed{1}$ 3 B ...	s_2	$(s_2, 1, s_0, B,\mathrm{R})$
...B B $\boxed{3}$ B ...	s_0	$(s_0, 3, s_1, 3,\mathrm{R})$
...B B 3 $\boxed{B}$ ...	s_1	no rule $\to$ stop