Solution5

Notation:

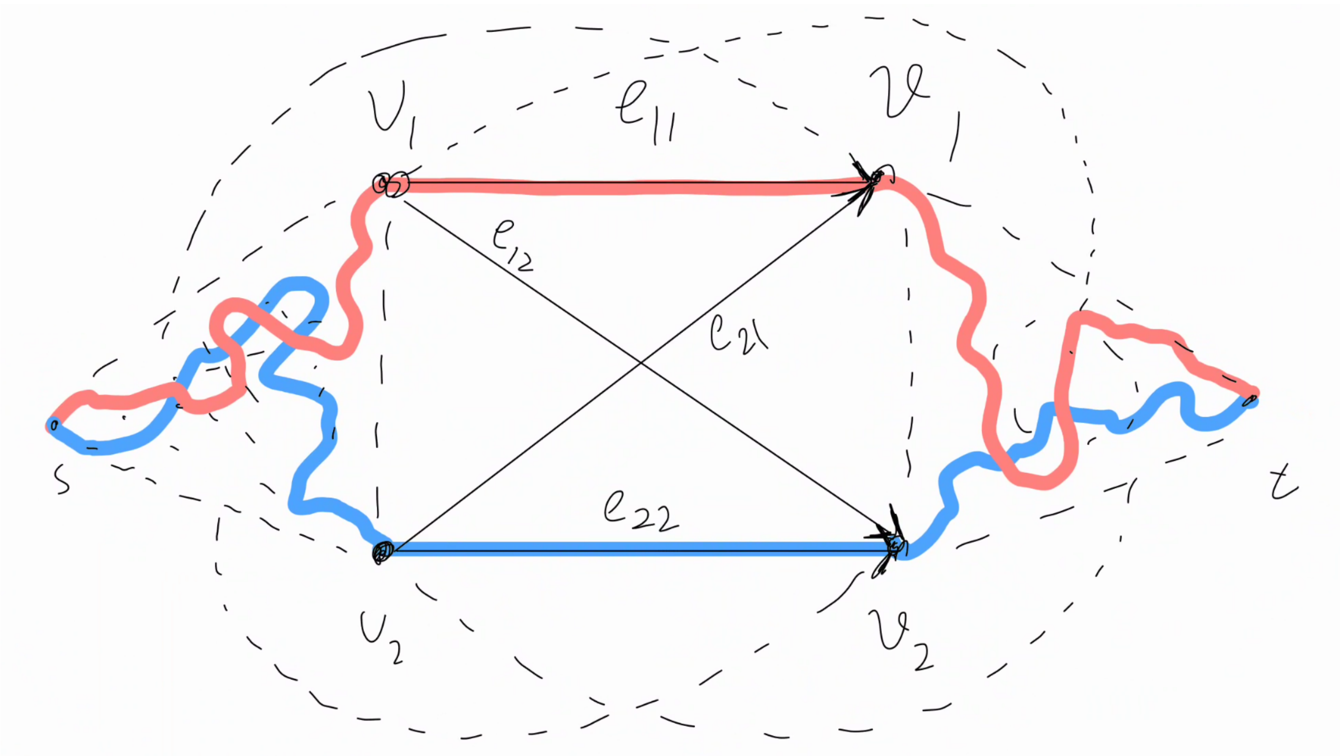
- => := this implies
- i.e := that is
- max(f) := maximum flow

In the network (G,s,t,c) max(f) = 24.
let: $e_{11} = u_1v_1$ $e_{22} = u_2v_2$ $e_{21} = u_2v_1$ $e_{12} = u_1v_2$

1:

> In the original graph there are 2 augmenting paths containing e_{11}, e_{22} respectively that can carry +32 flow if e_{11}/e_{22} weren't the bottleneck (i.e if we raise the capacity of e_{11}/e_{22} by 32 than you can flow +32 units each through those 2 augmenting paths)

Proof direct:
After raising the capacities c(e_{11})+=32 and c(e_{22})+=32 => max(f) = 88 = 24 + 32 + 32 => the e_{11} and e_{22} where a bottleneck of 2 augmenting paths that brought +32 flow each (and because they e_{11} and e_{22} are full even after raising their capacity they still may be a bottleneck) => in the original graph their exits 2 augmenting paths from s to t which can supply +32 flow each through the network and pass through e_{11}, e_{22} respectively.



2:

> e_{21} and e_{12} are full in the original graph.

Proof contradiction:
If they aren't then we can send more flow using the 2 augmenting paths above:

- for e_{21} we use the augmenting path s->u_2v_1->t
- for e_{12} we use the augmenting path s->u_1v_2->t
  But our flow was maximum. This is a contradiction.

Now if we increase the capacity of e_{21} to 31 => than we can flow +31 unis through the augmenting path s->u_2v_1->t (the part s->u_2 and v_1->t can now at least pump +1 more flow, but this wont have an effect in e_{12} cause when it could pump +32 it didn't have any effect in the first place (using 2)).
e_{12} is full (2) => if we decrease the capacity e_{12} by some number than also the max flow decreases (e_{12} is a bottleneck of <0,32>=+32) => when we decreased the capacity by 13 our flow also decreased the same amount (also the capacity in the first place was >=13).

This makes our max(f)=21+31-13=39

Klevis Bottleneck Theory :) (haha)

This is like a simple example and some semi precise definitions, just to get the idea of what I am thinking.
Note:

> If I could define define bottleneck precisely like:
> e is a bottleneck of value between <a,b> <=>
>
> - if raising c of e by n $\in$ [c(e),b] increases the max(f) by n.
> - if decreasing c of e by m $\in$ [a,c(e)] decreases the max(f) by m.
>   Notation: bottleneck(e)=neck(e)=<a,b>

Example:
neck(e_{11})=<0,+32>=neck(e_{22})=neck(e_{21})=neck(e_{12})=+32

Lemma:

> Lemma if e bottleneck <=> e full (in max f)

Theorem:

> The bottle neck edges are the edges going through the min cut.

and so on...