

## Paths and Circuits: Applications (continued)

### Combining RNA chains

UCGAGCUAGCGAAG

G-enzymes and UC-enzymes break the chain after each G and after each U, C, creating *complete enzyme digests*:

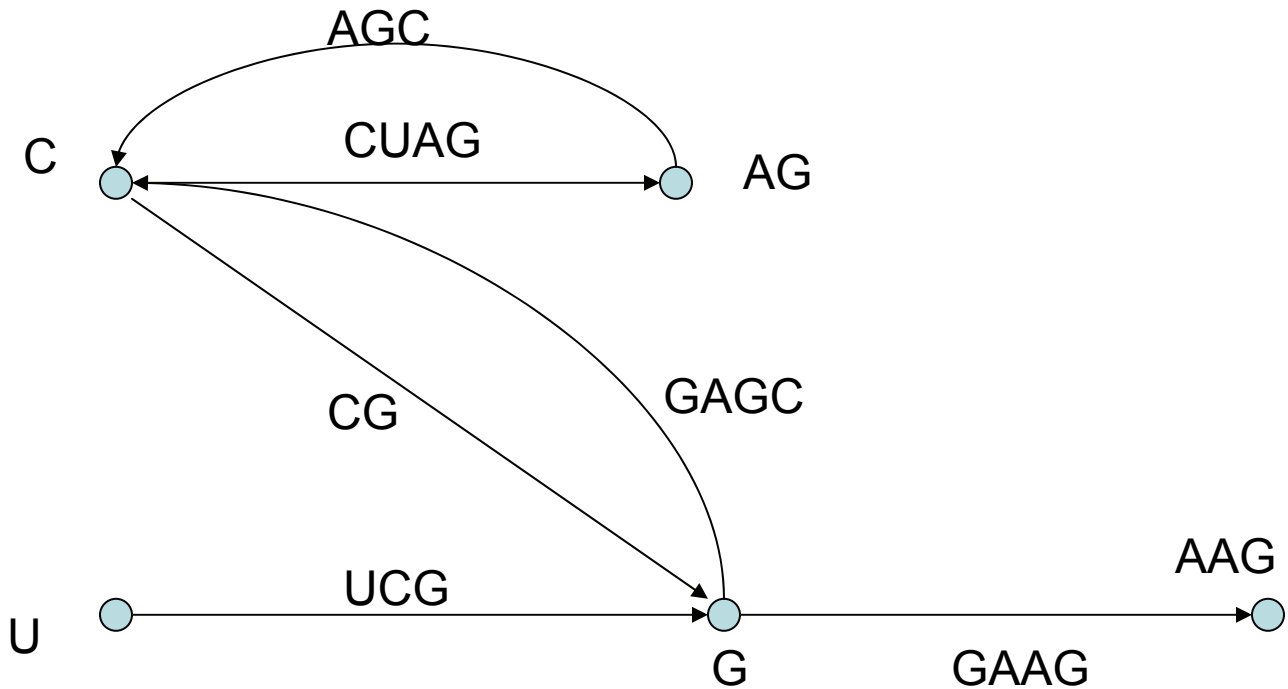
G → UCG AG CUAG CG AAG  
 UC → U C GAGC U AGC GAAG

One of the fragments is *abnormal* (there may be at most two abnormal fragments).

We can think of further breaking complete enzyme digest into *extended bases* by applying G and UC enzymes:

UCG	→	U, C, G
AG	→	AG
CUAG	→	C, U, AG
CG	→	C, G
AAG	→	AAG
U	→	U
C	→	C
GAGC	→	G, AG, C
AGC	→	AG, C
GAAG	→	G, AAG

Representing extended bases by vertices and representing complete enzyme digests by arcs, we can construct an oriented pseudograph:



Now the task of reconstruction is to find an Eulerian trail that ends with the abnormal fragment AAG. In the given example, there is only one possibility to put the arcs

UCG, CUAG, CG, GAGC, AGC, GAAG

in an Eulerian trail. It is

UCG – GAGC – CUAG – AGC – CG – GAAG

It remains to concatenate the arcs and remove the duplicates of vertices (each vertex appears twice). The result is the original chain:

UCGAGCUAGCGAAG

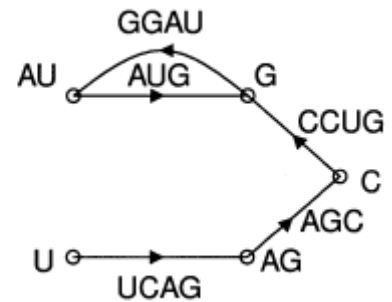
The problem 9 a on page 367 is solved similarly:

Find RNA chains with the given complete enzyme digests:

G-fragments: G, AUG, UCAG, CCUG  
 U,C-fragments: G, GGAU, C, C, U, AGC, U

*Solution:*

9. (a) The only abnormal fragment is G, so the chain ends with G. Since the abnormal fragment doesn't split, we look for Eulerian trails or circuits whose last vertex is G. The unique chain is UCAGCCUGGAUG.



## Scheduling

### Definitions:

A **directed network** is a digraph with an integer weight associated with each arc.

A **type I scheduling problem** is to find a shortest path (called in this case **critical path**) in a directed network where each arc represents a task, and vertices represent stages.

A **type II scheduling problem** is to find the length of the longest path in a directed network where each vertex represents a task, and arcs represent mutual dependencies of tasks.

The **slack** in a particular task is the time by which the task can be delayed without delaying the entire project.