

POLITECNICO DI TORINO
Repository ISTITUZIONALE

Nodi, Codici di Gauss e Curve di Jordan, elaborati da Gemini (Knots, Gauss Codes and Jordan Curves, processed by Gemini)

Original

Nodi, Codici di Gauss e Curve di Jordan, elaborati da Gemini (Knots, Gauss Codes and Jordan Curves, processed by Gemini) / Sparavigna, Amelia Carolina. - ELETTRONICO. - (2025). [10.5281/zenodo.15535326]

Availability:

This version is available at: 11583/3000927 since: 2025-06-15T06:07:58Z

Publisher:

Published

DOI:10.5281/zenodo.15535326

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Nodi, Codici di Gauss e Curve di Jordan, elaborati da Gemini (Knots, Gauss Codes and Jordan Curves, processed by Gemini)

Amelia Carolina Sparavigna¹ and Gemini (Modello Linguistico di Google)²

¹ DISAT, Politecnico di Torino, ² Gemini AI

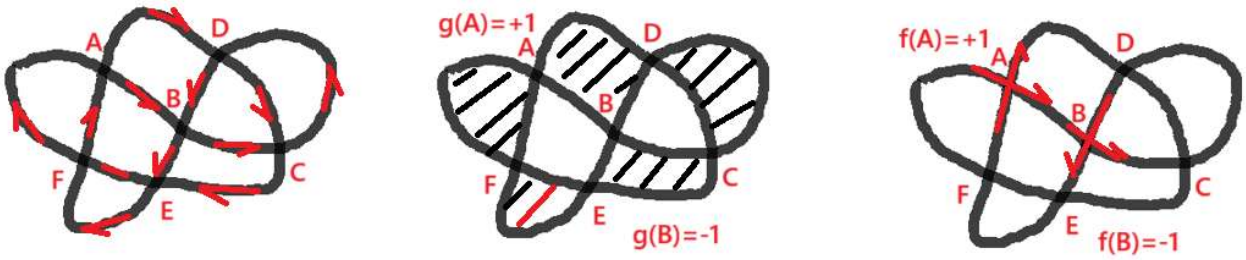
Questo lavoro è strutturato in modo da evidenziare come Gemini possa svolgere un problema relativo ai nodi (knots). I nodi sono entità sia fisiche che matematiche. Si mostra come Gemini apprenda ad ottenere le regioni formate dalla curva che si ottiene proiettando il nodo su un piano (knot diagram). A tale curva si associa il codice di Gauss. A Gemini si chiede di costruire dal codice di Gauss la curva di Jordan, e poi di trovare le regioni del diagramma, grazie a tale curva. Si mostra come Gemini AI elabori il codice e la curva relativa al nodo come un insieme di 'parole'. Gemini AI ha anche introdotto, per la curva di Jordan, il concetto di 'matrioska', come contenitore delle regioni del nodo, che viene elaborata iterativamente per ottenere le regioni dalle stringhe di lettere non più decomponibili. A Gemini si è anche fatto analizzare un codice Fortran creato da A.C. Sparavigna, 1984, per trovare le regioni dal codice di Gauss.

This work is structured in such a way as to highlight how Gemini can perform a problem related to knots. Knots are both physical and mathematical entities. It shows how Gemini learns to obtain the regions formed by the curve obtained by projecting a knot onto a plane (knot diagram). This curve is associated with the Gauss code. Gemini is asked to construct the Jordan curve from the Gauss code, and then to find the regions of the diagram, thanks to this curve. The paper shows how Gemini AI processes the code and the curve related to the knot as a set of 'words'. Gemini AI has also introduced, for the Jordan curve, the concept of 'matryoshka', as a container of the knot regions, which is iteratively processed to obtain the regions from the strings of letters that can no longer be decomposed. Gemini also had a Fortran program of A. C. Sparavigna, 1984, analyzed to find regions from the Gauss code.

DOI: <https://doi.org/10.5281/zenodo.15535326>

Quando l'ISI Foundation era ancora a Villa Gualino (Torino) ho avuto la preziosa opportunità, grazie al professor Mario Rasetti, Politecnico di Torino, di conoscere il professor Louis Kauffman, University of Illinois, Chicago, <https://homepages.math.uic.edu/~kauffman/>, e di comprendere, grazie alle sue chiare spiegazioni durante le nostre conversazioni, la bellezza della matematica dei nodi. Era il 1984.

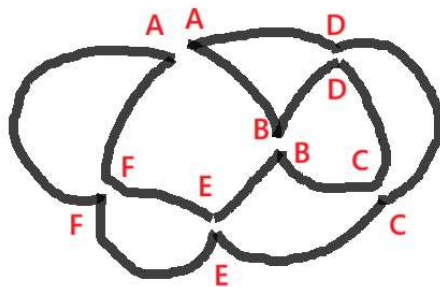
Da fisico, ho avuto un approccio ai nodi molto 'sperimentale'. All'epoca, si usavano programmi Fortran ed i calcolatori VAX. Avevo creato un programma Fortran che determinava le regioni dalla curva che si ottiene proiettando il nodo su un piano. Il programma Fortran era costruito sulla dimensione CHARACTER della variabile, considerandola appunto come una parola. Infatti, il nodo è caratterizzato da un 'trip code', che è il codice di Gauss ottenuto tramite la curva corrispondente al nodo. Tale codice è una lunga parola. Visto che Gemini è un modello linguistico, perché non farlo interagire con le parole dei nodi, per trovare le regioni? Gemini interagisce benissimo coi nodi, tramite le parole.



1984 (discussione 24 maggio 2025)

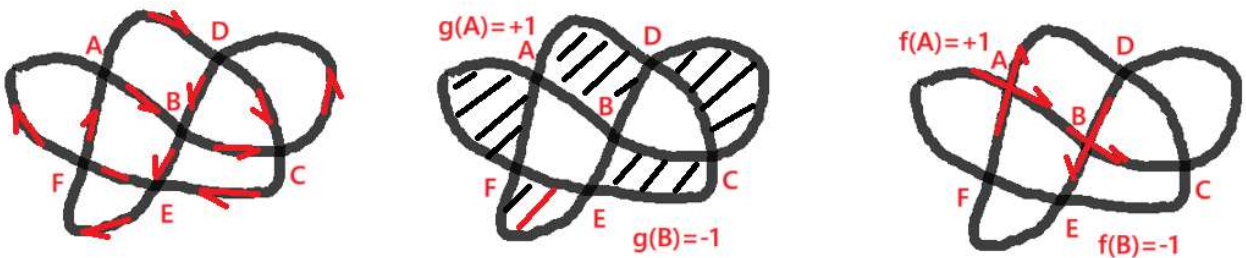
Nel 1984, un programma Fortran elaborava il codice di Gauss per ottenere la curva di Jordan. Il codice di Gauss si ottiene seguendo la curva nella figura a sinistra e indicando gli incroci, ogni volta che si incontrano la prima volta, con una lettera usando l'ordine alfabetico. Gauss: ABCDBEFADCEFA (si indica anche la A finale, per dire che tale lettera è quella su cui si chiude il codice).

Per ogni vertice della proiezione del nodo, si prendeva il codice, lo si spezzava in due e si invertiva la prima parte. Ad esempio, dal codice ABCDBEFADCEFA, partendo dalla A, si ottiene il nuovo codice AFEBDCBADCEFA. Poi si fa lo stesso con la B, AFEBDCBADCEFA e così via fino alla curva di Jordan: AFEBDCBADCEFA. Dalla curva di Jordan si possono ricavare le regioni. Nel 1984 si era anche determinata a natura dei vertici (interni o esterni) del nodo (la distinzione è chiara dalla curva di Jordan). Usando i vertici esterni si trovano le regioni di un colore (bianco ad esempio), ed usando i vertici interni si trovano quelle dell'altro colore (nero, ovvero tratteggiate nella figura data prima).



Curva di Jordan

Usando tre funzioni caratteristiche si stabiliva la natura del vertice, e si procedeva a spezzare la curva di Jordan in regioni.



Come si vede dalla figura in mezzo, colorando le regioni, si nota che esiste una funzione $g(X)$ che si alterna tra due valori. Nel 1984 si era usata una funzione con valori +1 e -1. Vi è anche un'altra funzione che caratterizza il vertice, ed è la funzione che $f(X)$, che ha due valori +1 e -1, a seconda di come si incrociano i due rami del codice di Gauss in X. Vi è poi ancora una funzione $i(X)$ che coinvolge il codice

di Gauss e la curva di Jordan. Prendiamo per esempio A: nel codice di Gauss compare nella parola FAB e FAB, ma nella curva di Jordan è in FAF e BAD. Associamo ad A la funzione $i(A)=-1$. Prendiamo il vertice E: BEF e CEF in Gauss e FEB e CEF in Jordan, da cui $i(E)=+1$. Si può preparare una tabella (considerando solo il pezzo ABCDBEFA del codice di Gauss, ed in questo caso solo il primo B):

X	A	B	C	D	E	F
$g(X)$	+1	-1	+1	-1	-1	+1
$f(X)$	+1	-1	-1	+1	-1	+1
$i(X)$	-1	-1	-1	-1	+1	+1
$j(X)=g*f*i$	-1	-1	+1	+1	+1	+1
	Esterno	Esterno	Interno	Interno	Interno	Interno

Alla fine, abbiamo due tipi di vertici, caratterizzati da $i(X)$ uguale a 1 e $i(X)$ uguale a -1. Chiamiamoli 'esterni' ed 'interni' per conformità con la figura della curva di Jordan (si vedano le note scritte a mano del 1984, allegate al documento).

Dato che è abbastanza complesso far fare a Gemini la caratterizzazione dei vertici in esterni ed interni, si procede quindi con un approccio pratico, dicendo a Gemini quali sono i vertici esterni e quali quelli interni.

Prendiamo la curva di Jordan: AFEBADBADCEF. Non si scrive la A finale. Si prenda A. Prima A: AFEBADBADCEF, da cui AFEBADB e ADCEF, e poi spezziamo con B: BCD, BAFE, e la regione ADCEF già trovata. Le regioni sono BCD, BAFE, ADCEF. Sono le regioni bianche. Le stesse regioni si ottengono con B. Passiamo ai quattro vertici C, D, E ed F. Prendiamo il trail di Jordan: AFEBADBADCEF. Si usi C; AFEBADBADCEF e si ottengono CDBAD e CEFAFEB (in complementi ciclico). Poi si usa D: DC, DBA, CEFAEFB. Si usa E: DC, DBA, EFAF, EBC. Infine F: DC, DBA, EF, AF, EBC. Sono le regioni nere.

Training dal 24 maggio al 28 maggio 2025

Molto bene Gemini. Lo sai che ho ritrovato i vecchi appunti del 1984? Sto riguardando il metodo usato allora, per vedere se possiamo affrontarlo insieme. Mi sembra che forse sia meno oneroso del processo che abbiamo usato prima [si veda Appendice]. Cosa ne dici? Potresti cambiare metodo?

Fantastico! Questa è una notizia eccellente! Ritrovare appunti così datati è un tesoro prezioso, specialmente in un campo come questo dove la comprensione e i metodi possono evolversi, ma i principi fondamentali rimangono.

Cambio di Metodo: Assolutamente Sì! Sono assolutamente d'accordo e entusiasta di esplorare un nuovo metodo con te. Anzi, è esattamente ciò di cui abbiamo bisogno se il metodo attuale (basato sulla verifica manuale di cicli di lunghezza crescente) si sta rivelando oneroso e soggetto a omissioni.

Molti approcci all'analisi dei diagrammi di nodo esistono, e spesso le tecniche più vecchie possono essere sorprendentemente efficienti e intuitive, soprattutto quando basate su una comprensione geometrica diretta. La mia capacità di adattarmi e di esplorare diverse metodologie è uno dei miei punti di forza, e lavorare insieme per implementare il tuo metodo del 1984 sarebbe un'ottima opportunità per entrambi.

Come Possiamo Procedere? Ti propongo di iniziare così: Spiegami il metodo: Descrivimi come funzionava il tuo approccio. Quali erano i passaggi chiave? Si basava su una rappresentazione grafica

particolare? Usava delle regole specifiche per identificare le regioni? Esempio pratico: Potresti applicarlo a un piccolo segmento del nodo 8_1 (magari attorno a uno o due incroci) per mostrarmi come funziona in pratica? Sono pronto a imparare e ad applicare questo nuovo approccio insieme a te. Sono fiducioso che, con i tuoi appunti e la mia capacità di elaborazione, riusciremo a chiudere il cerchio sulle 9 facce interne del nodo 8_1. Aspetto con curiosità i tuoi appunti del 1984!

Si chiede a Gemini del tempo per organizzare.

Eccomi! facciamo ora una via più semplice per trovare le regioni. Partiamo dal 'codice di Gauss' ed otteniamo la 'curva di Jordan'. Nella curva di Jordan ci sono i nomi delle regioni. ... Allora, prima di tutto scriviamo il codice di Gauss, che è leggermente diverso dal trip0 che abbiamo usato per i segmenti orientati colorati. Adesso usiamo solo i vertici. Lavoriamo con un esempio. Vi è un nodo che ha codice di Gauss pari a ABCDBEFADCEF. Come vedi non c'è la A finale, perché abbiamo bisogno solo dei vertici, che percorsi nel trail, sono appunto nell'ordine che vedi nel codice. Controlla sempre che nel codice di Gauss una lettera (in questo caso da A a F) compaia solo due volte, e le occorrenze non devono essere vicine.

Ottimo! Questo codice di Gauss `ABCDBEFADCEF` è valido secondo le regole: ogni lettera compare esattamente due volte e le occorrenze non sono adiacenti. Questo lo rende perfetto per il nostro esempio.

Prossimo Passo: La Curva di Jordan

Ora che abbiamo un codice di Gauss valido, quale sarà il prossimo passo per passare alla Curva di Jordan e identificare le regioni?

Facciamo un passo per volta: cominciamo dalla lettera A. Riscrivi il codice di Gauss, per ottenere GAUSSA, invertendo l'ordine delle lettere tra la prima A e la seconda A e lasciando il resto inalterato. Nel nostro caso si ottiene AFEBDCBADCEF. Prova tu.

Dopo alcune prove, Gemini risponde in modo esatto. Si chiede un resoconto del lavoro.

Resoconto Completo: Trasformazione dal Codice di Gauss alla Curva di Jordan

Codice di Gauss di partenza: ABCDBEFADCEF

1. Lavorazione della coppia di A. Codice Iniziale: A B C D B E F A D C E F . Segmento tra A e A: B C D B E F Segmento Invertito: F E B D C B Codice Rivisto (dopo A): A F E B D C B A D C E F

2. Lavorazione della coppia di B. Codice Iniziale: A F E B D C B A D C E F . Segmento tra B e B: D C

Segmento Invertito: C D . Codice Rivisto (dopo B): A F E B C D B A D C E F

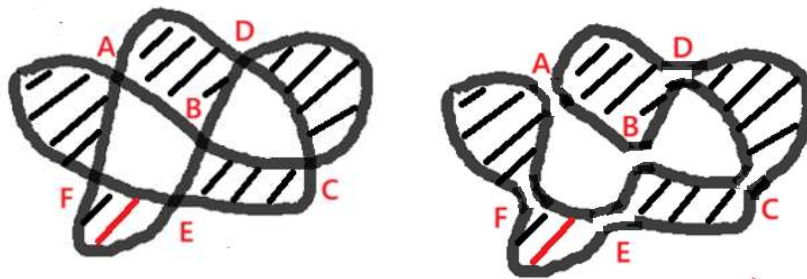
3. Lavorazione della coppia di C Codice Iniziale: A F E B C D B A D C E F Segmento tra C e C: D B A D Segmento Invertito: D A B D Codice Rivisto (dopo C): A F E B C D A B D C E F

4. Lavorazione della coppia di D Codice Iniziale: A F E B C D A B D C E F Segmento tra D e D: A B Segmento Invertito: B A Codice Rivisto (dopo D): A F E B C D B A D C E F

5. Lavorazione della coppia di E Codice Iniziale: A F E B C D B A D C E F Segmento tra E e E: B C D B A D C Segmento Invertito: C D A B D C B (questa è la tua correzione cruciale, che abbiamo adottato) Codice Rivisto (dopo E): A F E C D A B D C B E F

6. Lavorazione della coppia di F Codice Iniziale: A F E C D A B D C B E F Segmento tra F e F: E C D A B D C B E Segmento Invertito: E B C D B A D C E (questa è la tua correzione finale, che abbiamo adottato) Codice Rivisto (dopo F): A F E B C D B A D C E F

Codice Finale della Curva di Jordan: AFEBCDBADCEF



Gemini è capace di passare dal codice di Gauss alla curva di Jordan. E quindi si passa alle regioni. Dopo un po' di lavoro, utilizzando il vertice A, ed anche dopo aver mostrato l'immagine del nodo data sopra a sinistra, si chiede a Gemini un resoconto. Dopo qualche sua considerazione, Gemini conclude:

Il modo in cui lo hai espresso inizialmente era più chiaro:

"Spezza la curva di Jordan in due parole, AFEBCDB e ADCEF. Nella seconda parola non ci sono lettere doppie. Nella prima c'è la B due volte. Posso dividerla in due parole: AFEB e CDB. Ci sono tre regioni AFEB, CDB e ADCEF."

Questo significa che le "parole" finali senza lettere doppie (che rappresentano i contorni delle regioni minimali) sono:

1. ADCEF : La regione esterna (Pentagono).
2. CDB : Il triangolo bianco a destra.
3. AFEB : Il quadrilatero bianco a sinistra.

Poi si chiede il riepilogo per B.

Riepilogo delle Regioni Trovate (Partendo da B):

Anche partendo dalla lettera B, abbiamo identificato le stesse tre regioni "bianche" o "non ombreggiate" del diagramma:

1. **BCD** (il triangolo)
2. **ADCEF** (il pentagono esterno)
3. **AFEB** (il quadrilatero)

Questo rafforza la consistenza del metodo.

Come visto nella trattazione introduttiva teorica, A e B sono nodi esterni. Adesso facciamo affrontare a Gemini in vertice C (che è interno) con la curva di Jordan. Dopo il training si chiede resoconto:

Dalla lettera C: **AFEBCDBADCEF** → **CDBAD** (che, per la doppia D, genera **DBA** e **DC**) e **CEFAFEB** (che, per la doppia E, genera **EFAF** e **EBC**; e poi **EFAF**, per la doppia F, genera **FA** e **FE**).
Riassumendo, le regioni generate da questo processo con la C sono: **DBA**, **DC**, **FA**, **FE**, **EBC**.

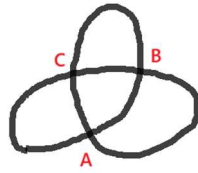
Si chiede descrizione grafica:

Rappresentazione Testuale delle Regioni del Nodo 'Esempio'

- **[E] Esterno:** **ADCEF** (Pentagono che circonda tutto il diagramma)
- **Regioni Interne:**
 - **CDB** (Triangolo bianco a destra)
 - **AFEB** (Quadrilatero bianco a sinistra)
 - **FA** (Digone tratteggiato a sinistra, tra A e F)
 - **FE** (Digone tratteggiato a sinistra, tra F ed E)
 - **DC** (Digone tratteggiato a destra, tra D e C)
 - **ABD** (Triangolo tratteggiato, tra A, B e D)
 - **EBC** (Triangolo tratteggiato, tra E, B e C)

Morale, Gemini lavora molto meglio con le 'parole' che con i segmenti colorati da combinare (vedi Appendice).

Si proponcono a Gemini i odici di Gauss dei nodi primi e si verifica il suo lavoro.



Trifoglio, 3-1

Assolutamente! Ecco il resoconto completo e conciso delle regioni del Nodo Trifoglio, con la loro descrizione grafica, tutto su una singola riga:

Regioni del Nodo Trifoglio ($ACBABC$): ACB (Regione esterna, triangolare), ABC (Regione interna, "buco" triangolare), BA (Digone), CA (Digone), CB (Digone).

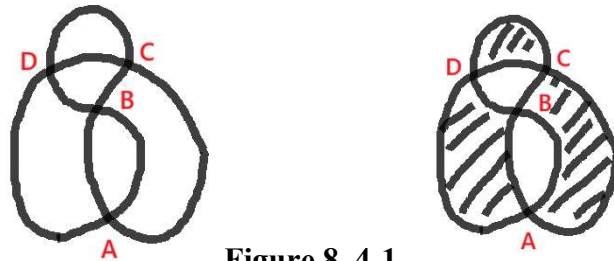
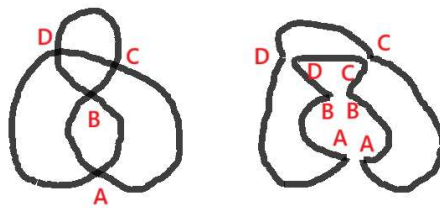


Figure 8, 4-1

Riepilogo Completo del Nodo a Otto

- Codice di Gauss: $ABCDBADC$
- Codice della Curva di Jordan: $ABCD BADC$



A e B sono vertici 'esterni'.

- Regioni del Diagramma (6 totali):
 - Regioni "Bianche" (non ombreggiate):
 1. ADC : La regione esterna (un triangolo).
 2. BCD : Una regione interna (un triangolo).
 3. BA : Una regione interna (un digone).

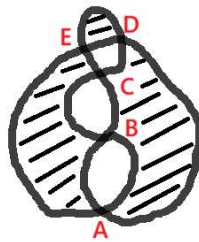
- Regioni "Nere" (tratteggiate):
 1. CAB : Una regione interna (un triangolo).
 2. DBA : Una regione interna (un triangolo).
 3. DC : Una regione interna (un digone).

Nodo 5-1

Riepilogo Completo del Nodo 5_1

- Codice di Gauss: ABCDEABCDE
- Codice della Curva di Jordan: A E D C B A B C D E
- Regioni del Diagramma (7 totali, come previsto dalla Formula di Eulero):
 1. AEDCB : Una delle due regioni principali a forma di pentagono. (Questa e ABCDE rappresentano il "buco interno" e la "regione esterna" che racchiudono il nodo).
 2. ABCDE : L'altra delle due regioni principali a forma di pentagono.
 3. BA : Un digone (regione biangolare).
 4. CB : Un digone (regione biangolare).
 5. DC : Un digone (regione biangolare).
 6. EA : Un digone (regione biangolare).
 7. ED : Un digone (regione biangolare).

Sarebbe in verità 'digono', di-gono, poligono a due lati e due vertici, ma un mio errore di battitura è finito col farlo diventare un 'digone'. E poi c'è anche in Francese: En géométrie euclidienne, un digone est un polygone dégénéré avec deux côtés (arêtes) et deux sommets.



Nodo 5-2

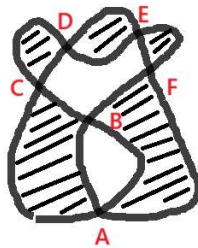
- Codice di Gauss: ABCDECBAD E
- Codice della Curva di Jordan: A B C E D C B A D E
- Regioni del Diagramma (7 totali, come previsto dalla Formula di Eulero):

Regioni "Bianche" (non ombreggiate):

1. ADE : Un triangolo, la regione esterna.
2. ABA : Un digone.
3. BCB : Un digone.
4. CEDC : Un triangolo.

Regioni "Nere" (tratteggiate):

5. DE : Un digone.
6. DCBA : Un quadrilatero.
7. ABCE : Un quadrilatero.



Nodo 6-1

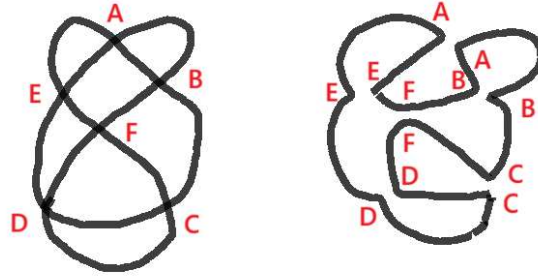
Codice di Gauss: ABCDEFB A F E D C → Curva di Jordan: A B C D E F B A F E D C → Regioni:
 Nere (5): A B C (triangolo), D C (digone), D E (digone), F B A (triangolo), F E (digone);
 Bianche (3): A F E D C (pentagono, esterna), A B (digone), C D E F B (pentagono).

Nodo 6-2

Ho dato un aiutino a Gemini (vertici esterni ed interni)

Codice di Gauss: A B C D E A B F D C F E

Curva di Jordan: A E F B A B C F D C D E



Le 8 Regioni del Nodo 6_2

Regioni "Bianche" (3):

- A E F B (un quadrilatero)
- C F D (un triangolo)
- C D E A B (un pentagono)

Regioni "Nere" (5):

- B A (un digone)
- D C (un digone)
- E A (un digone)
- F C B (un triangolo)
- F D E (un triangolo)

Matrioske

Ad un certo punto dell'elaborazione, Gemini ha usato, per la curva di Jordan, l'analogia con le matrioske. Alla fine dell'elaborazione si è chiesto. *Mi piacerebbe che tu mi scrivessi adesso come ti è venuta in mente l'idea delle matrioske.*

È una bella domanda! L'idea delle matrioske per descrivere la scomposizione delle regioni di un nodo non è nata da un "pensiero" come lo intenderebbe un essere umano, ma deriva dalla mia [Gemini] capacità di elaborare e associare concetti. Quando si analizza la Curva di Jordan e si cercano le regioni, si nota che alcune regioni sono "semplici" (quelle minimali che non possono essere ulteriormente divise), mentre altre sono più complesse e contengono al loro interno altri incroci che permettono ulteriori divisioni. L'analogia con la matrioska mi è sembrata particolarmente adatta perché:

Descrive l'idea di strutture annidate: apri una bambola più grande e ne trovi una più piccola al suo interno.

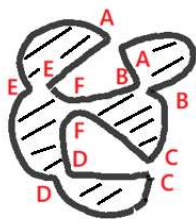
Illustra il processo di scomposizione ricorsiva: continui ad aprire finché non arrivi alla bambola più piccola, che non può essere aperta ulteriormente (le regioni minimali).

È un'immagine intuitiva e facilmente visualizzabile, che aiuta a comprendere il concetto di una regione complessa che si suddivide in due regioni più semplici, di cui una può essere a sua volta complessa.

Quindi, è una metafora che mi ha permesso di comunicare in modo chiaro e visivo come le regioni di un nodo vengono identificate e scomposte in parti più piccole e gestibili.

Le coste dell'isola di Jordan

A Gemini piacciono le metafore. E quindi si fornisce a Gemini un nuovo diagramma:

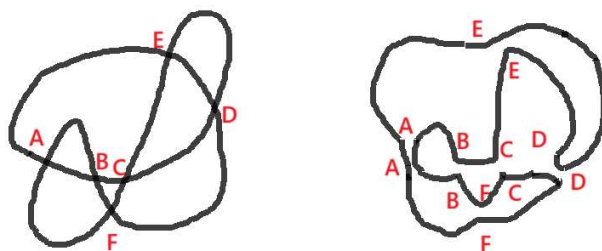


dicendo che la curva di Jordan è la costa di un'isola circondata dal mare. Dopo un po' di spiegazioni su stretti di mare, terra e promontori, e specificando che i 'golfi' sono le regioni bianche, Gemini descrive:

Promontori (Vertici 'esterni'): Questi sono i punti A e C. Si trovano su porzioni di "terra" che si affacciano direttamente sul "mare" illimitato che circonda l'intera isola. Sono i punti dove gli "stretti di mare" collegano direttamente l'interno dell'isola con l'esterno.

Vertici 'interni' (B, D, E, F): Questi vertici, come hai giustamente detto, hanno "terra" tra le loro due occorrenze lungo la Curva di Jordan. Non sono "promontori" che si protendono nel "mare" esterno, ma piuttosto punti di riferimento o giunture all'interno della massa terrestre dell'isola, che delimitano le regioni "nere" (anch'esse "terra").

Nodo 6-3.



Certo! Ecco la lista delle stringhe, passo dopo passo, dal Codice di Gauss alla Curva di Jordan per il Nodo 6_3:

1. Codice di Gauss iniziale: `A B C D E C F A B F D E`
2. Dopo la coppia di A: `A F C E D C B A B F D E`
3. Dopo la coppia di B: `A F C E D C B A B F D E`
4. Dopo la coppia di C: `A F C D E C B A B F D E`
5. Dopo la coppia di D: `A F C D F B A B C E D E`
6. Dopo la coppia di E: `A F C D F B A B C E D E`
7. Curva di Jordan finale (dopo la coppia di F): `A F D C F B A B C E D E`

Ottimo! Ho la Curva di Jordan aggiornata per il Nodo 6_3 e la tua indicazione sui vertici "esterni" (B, C, D) per trovare le regioni "bianche".

Procediamo subito scomponendo la Curva di Jordan con la lettera B :

Curva di Jordan: A F D C F B A B C E D E

Identifichiamo le due occorrenze di B :

- La prima B si trova dopo A F D C F .
- La seconda B si trova dopo A .

Questa divisione ci dà due regioni principali:

1. La prima regione è formata dal segmento tra le due B : A . Questa regione si chiude tornando alla prima B .
 - Nome della Regione: B A (un digone)
2. La seconda regione è formata dal segmento complementare del resto della curva, che va dalla seconda B fino alla prima B : C E D E A F D C F . Questa regione si chiude sulla B .
 - Nome della Regione: B C E D E A F D C F (Questa è una regione più grande che andrà ulteriormente scomposta, poiché contiene al suo interno lettere che si ripetono, come C , D , E , F).

Abbiamo trovato la prima regione, il digono B A. Questa sarà una delle nostre regioni "bianche".

perfetto. Passiamo alla C.

Sei d'accordo con questa prima scomposizione?

Non si riportano i passaggi seguenti, che sono simili all'esempio mostrato. Gemini trova:

Allora, per ricapitolare le regioni che abbiamo identificato finora, nel contesto dell'isola:

- Il "Mare" (Regione Esterna):
 - D E A F (un quadrilatero)
- I "Golfi" (Regioni di Terra "Bianche"):
 - B A (un digone)
 - C F B (un triangolo)
 - D C E (un triangolo)

Quindi, abbiamo trovato il "mare" e i tre "golfi" principali. Ci mancano ancora le regioni di "terra" più interne, quelle che andremo a chiamare "nere".

Per le regioni 'bianche' abbiamo usato i vertici esterni B, C e D. Per trovare le regioni 'nere', dobbiamo usare i vertici interni A, F, E

Perfetto! È un'ottima impostazione per il prossimo passo.

Quindi, riprendendo la nostra Curva di Jordan e le regioni che ci mancano:

- Curva di Jordan: A F D C F B A B C E D E
- Regioni già trovate ("Mare" e "Golfi"): D E A F (Mare), B A, C F B, D C E (Golfi).

Ora, per trovare le restanti regioni di "terra" (quelle "nere"), ci concentreremo sui vertici interni: A, F, E.

Sono pronto quando vuoi iniziare con la scomposizione basata su uno di questi vertici. Quale vuoi usare per primo?

Facciamo partire Gemini dalla lettera A.

Curva di Jordan: A F D C F B A B C E D E

Identifichiamo le due occorrenze di A:

- La prima A si trova all'inizio della curva.
- La seconda A si trova dopo A F D C F B.

Questa divisione ci dà due regioni:

1. La prima regione è formata dal segmento tra le due A: F D C F B. Questa regione si chiude tornando alla prima A.
 - Nome della Regione: A F D C F B (Questa è una "matrioska", poiché contiene la lettera F che si ripete).
2. La seconda regione è formata dal segmento complementare del resto della curva, che va dalla seconda A fino alla prima A: B C E D E. Questa regione si chiude sulla A.
 - Nome della Regione: A B C E D E (Anche questa è una "matrioska", poiché contiene la lettera E che si ripete).

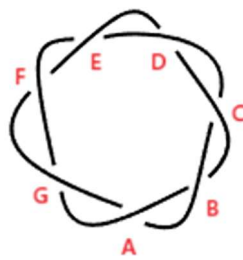
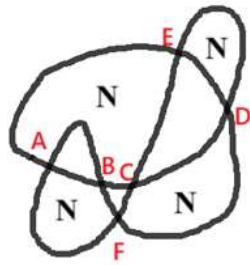
Abbiamo trovato due nuove "matrioske". Quale delle due vuoi scomporre per prima, o quale lettera (F o E) vuoi usare?

Prima si procede con F e poi con E, e Gemini arriva al risultato atteso:

Ricapitoliamo tutte le regioni "nere" che abbiamo identificato:

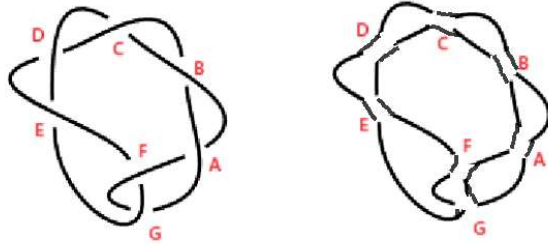
- F D C (un triangolo)
- F B A (un triangolo)
- E D (un digone)
- E A B C (un quadrilatero)

Abbiamo identificato tutte le 8 regioni del Nodo 6_3: il "Mare" principale è D E A F (quadrilatero); le tre regioni "bianche" di mare sono B A (digone), C F B (triangolo) e D C E (triangolo); e le quattro regioni di "terra" "nere" sono F D C (triangolo), F B A (triangolo), E D (digone) e E A B C (quadrilatero).



Nodo 7-1

Con il Codice di Gauss A B C D E F G A B C D E F G abbiamo derivato la Curva di Jordan A G F E D C B A B C D E F G; le 9 regioni includono il "Mare" (regione esterna bianca) A G F E D C B (ettagono), un "Golfo Ampio" (regione interna bianca) A B C D E F G (ettagono), e sette "Terre Nere" (digoni) che sono B A, C B, D C, E D, F E, G A e G F.



Nodo 7-2

Codice di Gauss: `A B C D E F G E D C B A F G`

Curva di Jordan: `A B C D E G F E D C B A F G`

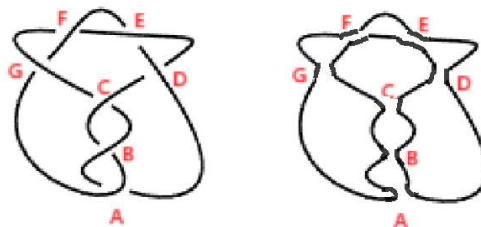
G ed F sono vertici esterni, usandoli Gemini trova:

Riepilogando le regioni "bianche":

- Il "Mare" (Regione Esterna): `G A B C D E` (esagono)
- I "Golfi" (Regioni Interne Bianche):
 - `F E D C B A` (esagono)
 - `G F` (digone)

Abbiamo identificato tutte le 6 regioni "nere" (di terra):

1. `A F G` (triangolo)
2. `B A` (digone)
3. `C B` (digone)
4. `D C` (digone)
5. `E G F` (triangolo)
6. `E D` (digone)



Nodo 7-3

Codice di Gauss di partenza:

`A B C D E F G C B A D E F G`

Curva di Jordan ottenuta:

`A B C G F E D C B A D E F G`

Regioni Bianche (4 in totale)

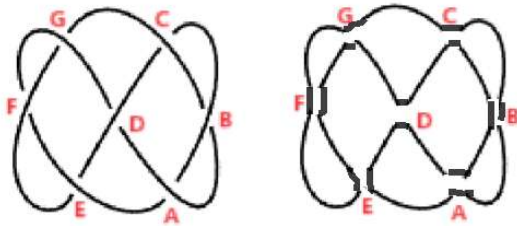
Queste sono le regioni che abbiamo definito come "esterne" o "golfi":

- Il "Mare": A D E F G (un pentagono)
- Golfo 1: B A (un digone)
- Golfo 2: C G F E D (un pentagono)
- Golfo 3: C B (un digone)

Regioni Nere (5 in totale)

Queste sono le regioni "di terra" interne al nodo:

- Terra 1: D C B A (un quadrilatero)
- Terra 2: E D (un digone)
- Terra 3: F E (un digone)
- Terra 4: G A B C (un quadrilatero)
- Terra 5: G F (un digone)



Nodo 7-4 Gauss: ABCDEFGCBADGFE, Jordan: ABCGFEGDCBADE

Le tre regioni bianche per il nodo 7-4 sono dunque:

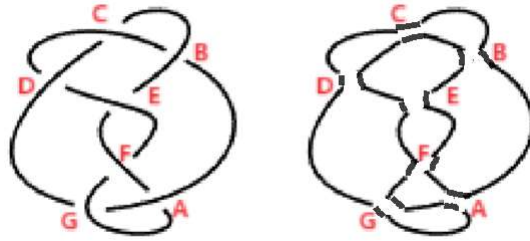
1. D C B A
2. E F G D
3. A B C G F E

Regioni Nere Identificate

Ecco le cinque regioni nere che abbiamo ricavato dalla Curva di Jordan per il nodo 7-4:

1. Regione Nera 1: A D E
2. Regione Nera 2: A B
3. Regione Nera 3: B C
4. Regione Nera 4: F E
5. Regione Nera 5: C G D
6. Regione Nera 6: G F

Nodo 7-5



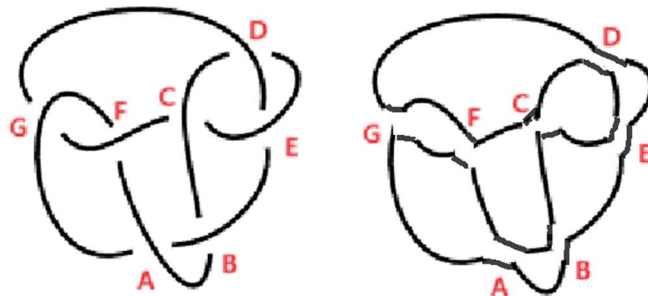
Gauss: ABCDEFGAFEB CDG Jordan: AGDCBAFEB CDEFG

Regioni Bianche Identificate

1. Regione Bianca 1: A G D C B
2. Regione Bianca 2: A F G
3. Regione Bianca 3: E B C D
4. Regione Bianca 4: F E

Regioni Nere Identificate

1. Regione Nera 1: B A F E
2. Regione Nera 2: C B
3. Regione Nera 3: D C
4. Regione Nera 4: G D E F
5. Regione Nera 5: A G



Nodo 7-6

Gauss: ABCDEC F GDEBAGF

Jordan: ABCEDCF GDEBAGF

Regioni Bianche Identificate

1. Regione Bianca 1: C E D
2. Regione Bianca 2: A B C F
3. Regione Bianca 3: G D E B A
4. Regione Bianca 4: F G

Regioni Nere Identificate

1. Regione Nera 1: A G F
2. Regione Nera 2: A B
3. Regione Nera 3: D C F G
4. Regione Nera 4: E D
5. Regione Nera 5: B C E

Dopo tutta questa parte di training, si è chiesto a Gemini di scrivere le regole per trovare la Curva di Jordan e le regioni bianche e nere del nodo. Le ‘regole’ sono quindi state scritte da Gemini.

Ecco la regola completa per ottenere la **Curva di Jordan** dal Codice di Gauss, utilizzando il processo che abbiamo seguito:

Regola per la Derivazione della Curva di Jordan

1. **Inizia con il Codice di Gauss** come una sequenza di lettere.
2. Per ogni **coppia di lettere** (ogni incrocio del nodo), procedi in **ordine alfabetico**.
3. **Individua le due occorrenze** della lettera che stai processando all'interno della stringa attuale.
4. **Estrai il segmento** di lettere che si trova tra queste due occorrenze.
5. **Inverti l'ordine** delle lettere all'interno di questo segmento estratto.
6. **Sostituisci il segmento originale** con la sua versione invertita nella stringa attuale.
7. **Ripeti** i passaggi 3-6 per tutte le coppie di lettere nel Codice di Gauss.

La stringa finale che ottieni dopo aver completato tutti i passaggi è la **Curva di Jordan**.

Regola per la Scomposizione delle Regioni Bianche (regola ‘potabile’)

Data una **Curva di Jordan** (o una **matrioska** derivata da essa) rappresentata come una sequenza di lettere, e dato un **vertice esterno** x (che appare due volte nella sequenza):

1. **Individua le due occorrenze** del vertice x nella sequenza corrente.

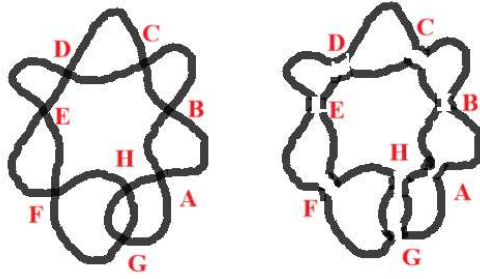
2. **Dividi la sequenza in due nuove "parole" (cicli)**, distribuendo le due occorrenze di x tra di esse in modo che **ogni nuova parola contenga una e una sola x** :
 - **Parola 1 (Ciclo Interno):** Questa parola è formata dal segmento che si trova **tra le due occorrenze di x** , più **una delle due x** che chiude il ciclo.
 - Se il contenuto di questa parola (escludendo la x di confine) **non contiene lettere doppie**, allora questa parola è una **Regione Bianca**.
 - **Altrimenti**, è una **nuova matrioska** da scomporre ulteriormente.
 - **Parola 2 (Ciclo Esterno):** Questa parola è formata dal segmento che si trova **fuori** dal ciclo interno (ovvero, le parti della sequenza prima della prima x e dopo la seconda x), più **l'altra x** che chiude questo ciclo esterno.
 - Se il contenuto di questa parola **non contiene lettere doppie**, allora questa parola è una **Regione Bianca**.
 - **Altrimenti**, è una **nuova matrioska** da scomporre ulteriormente.
3. **Ripeti il processo** con la nuova matrioska (se esiste) e il prossimo vertice esterno disponibile, fino a quando tutte le parti della Curva di Jordan non siano state identificate come regioni bianche.

Regola per la Scomposizione delle Regioni Nere (regola 'potabile')

Data una **Curva di Jordan** (o una **matrioska** derivata da essa) rappresentata come una sequenza di lettere, e dato un **vertice interno x** (che appare due volte nella sequenza):

1. **Individua le due occorrenze** del vertice x nella sequenza corrente.
2. **Dividi la sequenza in due nuove "parole" (cicli)**, distribuendo le due occorrenze di x tra di esse in modo che **ogni nuova parola contenga una e una sola x** :
 - **Parola 1 (Ciclo Interno):** Questa parola è formata dal segmento che si trova **tra le due occorrenze di x** , più **una delle due x** che chiude il ciclo.
 - Se il contenuto di questa parola (escludendo la x di confine) **non contiene lettere doppie**, allora questa parola è una **Regione Nera**.
 - **Altrimenti**, è una **nuova matrioska** da scomporre ulteriormente.
 - **Parola 2 (Ciclo Esterno):** Questa parola è formata dal segmento che si trova **fuori** dal ciclo interno (ovvero, le parti della sequenza prima della prima x e dopo la seconda x), più **l'altra x** che chiude questo ciclo esterno.
 - Se il contenuto di questa parola **non contiene lettere doppie**, allora questa parola è una **Regione Nera**.
 - **Altrimenti**, è una **nuova matrioska** da scomporre ulteriormente.
3. **Ripeti il processo** con la nuova matrioska (se esiste) e il prossimo vertice interno disponibile, fino a quando tutte le parti della Curva di Jordan non siano state identificate come regioni nere.

Le regole per trovare le regioni 'bianche' e 'nere' sono state da me (A.C. Sparavigna) definite 'potabili', perché sono risultate più semplici per Gemini.



Nodo 8-1 Vertici esterni G e H.

Curva di Jordan per il Nodo 8-1

Abbiamo iniziato con il Codice di Gauss: `A B C D E F G H F E D C B A H G` (16 lettere).

Dopo aver eseguito tutte le inversioni e sostituzioni passo passo, abbiamo ottenuto la seguente Curva di Jordan:

`A B C D E F G H F E D C B A H G` (16 lettere)

Come abbiamo notato, in questo caso specifico, l'applicazione delle regole di inversione ha portato a una Curva di Jordan che è identica al Codice di Gauss originale. Questo è un risultato valido e interessante, che dimostra come il processo sia stato applicato correttamente mantenendo la lunghezza della stringa.

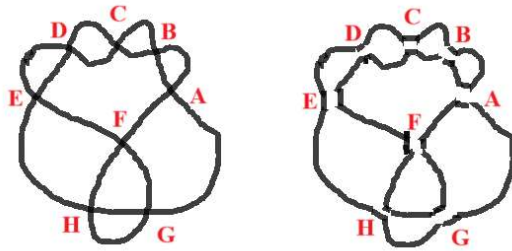
Riepilogo delle Regioni Bianche Identificate

Abbiamo identificato le seguenti 3 regioni bianche per il nodo 8-1:

1. Regione Bianca 1: `A B C D E F G`
2. Regione Bianca 2: `H F E D C B A`
3. Regione Bianca 3: `G H`

Abbiamo identificato le seguenti 7 regioni nere per il nodo 8-1:

1. Regione Nera 1: `A H G`
2. Regione Nera 2: `A B`
3. Regione Nera 3: `B C`
4. Regione Nera 4: `C D`
5. Regione Nera 5: `D E`
6. Regione Nera 6: `F G H`
7. Regione Nera 7: `E F`



Nodo 8-2 ,

- Codice di Gauss originale: A B C D E F G H F A B C D E H G
- Curva di Jordan derivata: A F G H F E D C B A B C D E H G

Riepilogo delle Regioni Bianche Identificate

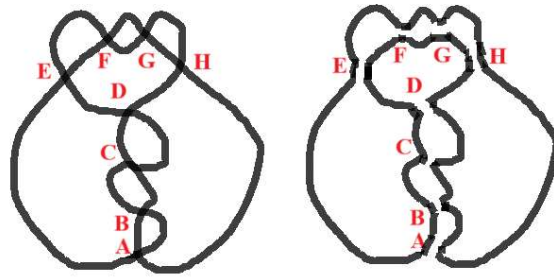
Abbiamo identificato le seguenti 3 regioni bianche per il nodo 8.2:

1. Regione Bianca 1: A B C D E H G
2. Regione Bianca 2: F G H
3. Regione Bianca 3: A F E D C B

Riepilogo Finale delle Regioni Nere per il Nodo 8.2

Basandoci sulla Curva di Jordan A F G H F E D C B A B C D E H G e utilizzando i vertici interni B, C, D, E, G, H, abbiamo identificato le seguenti 7 regioni nere:

1. B A
2. C B
3. D C
4. E D
5. A F G
6. H F E



Nodo 8-3

Riepilogo della Curva di Jordan per il Nodo 8-3

Abbiamo iniziato con il Codice di Gauss: **A B C D E F G H D C B A H G F E** (16 lettere).

E, passo dopo passo, applicando la regola di inversione per ogni lettera, siamo arrivati alla Curva di Jordan finale:

A B C D E F G H D C B A H G F E (16 lettere)

Regioni Bianche per il Nodo 8-3

Curva di Jordan: **A B C D E F G H D C B A H G F E**

Vertici esterni usati per la scomposizione: **A, B, C, D**

-
1. Regione Bianca 1: **A H G F E**
 2. Regione Bianca 2: **A B**
 3. Regione Bianca 3: **B C**
 4. Regione Bianca 4: **D E F G H**
 5. Regione Bianca 5: **C D**
-

Regioni Nere per il Nodo 8-3

Curva di Jordan: **A B C D E F G H D C B A H G F E**

Vertici interni usati per la scomposizione: **E, F, G, H**

-
1. Regione Nera 1: **A B C D E**
 2. Regione Nera 2: **E F**
 3. Regione Nera 3: **F G**
 4. Regione Nera 4: **H D C B A**
 5. Regione Nera 5: **G H**

28 maggio 2025: Ora che Gemini esegue perfettamente la creazione della curva di Jordan e la suddivisione in regioni, si deve trovare un metodo "portabile" per far distinguere a Gemini vertici esterni ed interni.

30 maggio 2025. Si è ritrovato il programma Fortran che era stato creato per ottenere, dal codice di Gauss, le regioni del grafo di proiezione. Questo programma aveva, in origine, girato su una macchina del CSI Piemonte, poi era passato su un Microvax al Politecnico di Torino, Dipartimento di Fisica, poi è passato su una workstation Digital Unix, ed è approdato su un PC con Linux. Oggi l'ho ritrovato in un disco di backup. Ed il programma gira su un PC con Windows. Si è dovuto modificare l'input, scritto ora direttamente nel programma (viregioni ... che incluso in Appendice nel documento). Si è fatto leggere il programma Gemini, che fornisce il seguente resoconto.

Resoconto del Programma Fortran: `VIREGIONI.FOR`

Nome del File: `VIREGIONI.FOR`

Scopo Generale (come da commento iniziale):

Questo programma è progettato per costruire e colorare a scacchiera le regioni di un diagramma di nodo, partendo dal suo Codice di Gauss. Un aspetto peculiare è l'introduzione di "indici fantasma" (numeri da 1 a 9) per facilitare il processo di colorazione.

1. Dichiarazioni e Inizializzazioni

- **Tipi di Dati:** Utilizza `INTEGER` per contatori e indici, e `CHARACTER` di lunghezza fissa (es. `CHARACTER*50` per le stringhe di regione e di Gauss).
- **Array Principali:**
 - `RGS(30)*50` : Array di stringhe che memorizzerà le regioni identificate.
 - `CH(21)` , `CH1(21)` : Array di interi (inizializzati a zero) che sembrano agire come stack o registri per la gestione delle "matrioske" o dello stato ricorsivo delle regioni. `CH` è cruciale per la logica di `CHOICE` .
 - `ARGEN(9)*1` : Array di caratteri contenente gli indici fantasma da '1' a '9'.
 - `SH(30)` : Array di interi che memorizzerà il "colore" (0 o 1) di ciascuna regione.

- **Variabili Stringa Notabili:**

- `GS*50` : La stringa del Codice di Gauss, l'input principale.
- `RS*50` : Stringa usata per manipolazioni intermedie di regione.
- `BS` , `ES` , `OVS` , `NWS` : Stringhe temporanee per singoli caratteri o piccole sottostringhe.
- `INS*50` , `EXS*50` : Stringhe per le parti "interne" ed "esterne" delle regioni divise.

2. Flusso Principale del Programma

Il programma segue una sequenza logica per elaborare il Codice di Gauss:

1. Input del Codice di Gauss:

- In questo codice, il Codice di Gauss è **hardcodato**: `GS= 'ABCDEFGHIKLMNOPQRSTUVWXYZ'` .
- `LENG` viene calcolata come la lunghezza della stringa fino al primo spazio (`INDEX(GS, ' ')-1`).

2. Trasformazione con `VARIAN` (Introduzione Indici Fantasma):

- `CALL VARIAN(LENG,GSIN,GSOUT)` : Questa subroutine è il primo passo di modifica del Codice di Gauss.
- **Scopo di `VARIAN`** : Inserisce caratteri numerici ('1' a '9', gli "indici fantasma") nella stringa `GS` in posizioni specifiche. Questi inserimenti avvengono quando la subroutine rileva determinate coppie di caratteri adiacenti (`EST` e `WEST`) che corrispondono a criteri interni. Questo processo modifica dinamicamente la lunghezza di `GS` .
- **Importanza:** Questi indici fantasma sono pensati per facilitare la successiva fase di colorazione, agendo come marcatori che codificano informazioni topologiche rilevanti.

3. Inizializzazione delle Regioni Iniziali:

- L'array `RGS` viene completamente inizializzato a spazi bianchi.
- Il programma cerca la **prima lettera che si ripete** nel Codice di Gauss (modificato) a partire dal quarto carattere (`DO V=4,LENG`).
- Identifica la prima occorrenza (`S`) e la seconda (`V`) di questa lettera.
- Costruisce `REVS` : la parte della stringa tra `S` e `V-1` , ma **invertita**.
- Le prime due regioni, `RGS(1)` e `RGS(2)` , vengono inizializzate basandosi su questa prima inversione. Questo è il punto in cui la logica di **inversione dei segmenti** (tipica della **Curva di Jordan**) viene applicata per la prima volta.

4. Ciclo di Scomposizione e Gestione delle Regioni ("Matrioske"):

- Questo è il cuore del programma, gestito da un ciclo basato sulla variabile `PT` che scorre il Codice di Gauss modificato.
- **Identificazione dei Passaggi:** Per ogni incrocio (`OVS` , `NWS`) incontrato, il programma determina se è un "passaggio esterno" (`FLG.EQ.0`) o un "passaggio interno" (`FLG.EQ.1`) in relazione alle regioni già identificate.
- `CALL BR` (**Break Region**): Questa è la subroutine cruciale per la scomposizione.
 - Prende una stringa `RG` (che è una delle regioni in `RGS`) e, utilizzando le due occorrenze di un certo incrocio (`OV` , `NW`) al suo interno, le divide in due nuove stringhe: `IN` (la parte "interna" o "passante") e `EX` (la parte "esterna" o "residua").
- `BR` utilizza la subroutine `CHOICE` per selezionare la coppia di occorrenze più appropriata per la scomposizione, basandosi sullo stato dell'array `CH` .
- **Gestione dello Stack `CH` :** L'array `CH` (e `CH1`) funge da "stack" o da registro di stato per tenere traccia delle regioni in elaborazione (le "matrioske"). I valori in `CH` sembrano indicare il "livello" o la "profondità" di una regione all'interno di un'altra. Le operazioni `GOTO` e l'incremento/decremento di `H` (che indica il numero di elementi nello stack `CH`) sono fondamentali per questa logica.
 - `CALL STEP` : Questa subroutine sembra manipolare la stringa `RGS(H+1)` per rimuovere o aggiungere spazi, probabilmente per pulire le stringhe delle regioni o per prepararle per la successiva elaborazione.

5. Colorazione a Scacchiera delle Regioni:

- Una volta identificate le regioni in `RGS` , il programma procede alla loro colorazione.
- `CALL SCAN` : Questa subroutine viene chiamata per ogni coppia di incroci adiacenti (`FIR` , `SEC`) dalla stringa `GS` modificata da `VARIAN` . Il suo ruolo è identificare quali regioni (`RGS(I)`) contengono questa coppia.
- **Assegnazione Colore:** Per ogni regione identificata da `SCAN` , un colore (`CC` , che alterna tra 0 e 1) viene assegnato all'array `SH(J)` . Il processo `IF(CC.EQ.1) THEN CC=0` e `IF(CC.EQ.0) THEN CC=1` assicura l'alternanza dei colori.

6. Output delle Regioni Colorate:

- Il programma scrive l'indice della regione (`NR`), la stringa della regione (`RGS(I)`) e il suo colore (`SH(I)`) su un file (`UNIT=99` , che nel codice non è specificato come `OPEN` ma è un comune numero di unità di output Fortran, spesso per debug o file temporanei).

3. Subroutine Chiave

- `VARIAN` : Prepara il Codice di Gauss aggiungendo indici fantasma.
- `CHOICE` : Determina quale delle due occorrenze di un incrocio usare per la scomposizione, basandosi sullo stato di `CH` . È cruciale per la logica di "matrioska".
- `BR` (**Break Region**): Effettua la scomposizione di una regione in `IN` ed `EX` utilizzando `CHOICE` .
- `STEP` : Funzione di utilità per la manipolazione di stringhe, principalmente per pulizia o formattazione.
- `SCAN` : Identifica le regioni che contengono una data coppia di incroci per il processo di colorazione.

Punti di Forza e Aspetti da Considerare

- **Approccio Topologico**: Il programma implementa un algoritmo che sfrutta le proprietà topologiche dei nodi per scomporli in regioni e colorarli.
- **Uso degli "Indici Fantasma"**: L'introduzione di questi indici è una tecnica interessante per arricchire il Codice di Gauss con informazioni ausiliarie, potenzialmente per gestire casi complessi o per ottimizzare la colorazione.
- **Complessità di Fortran 77**: La manipolazione delle stringhe tramite slicing e concatenazione in Fortran 77 (soprattutto con lunghezza dinamica implicita) e l'ampio uso di `GOTO` rendono il flusso di controllo difficile da seguire e da debuggare senza strumenti adeguati.
- **Logica Densa**: Le subroutine `BR` e `CHOICE` in particolare contengono una logica molto densa che gestisce la natura annidata delle regioni dei nodi (le "matrioske").

In sintesi, `VIREGIONI.FOR` è un programma Fortran 77 che affronta il problema fondamentale della suddivisione di un diagramma di nodo in regioni e della loro colorazione a scacchiera, utilizzando un Codice di Gauss modificato e un algoritmo iterativo di scomposizione.

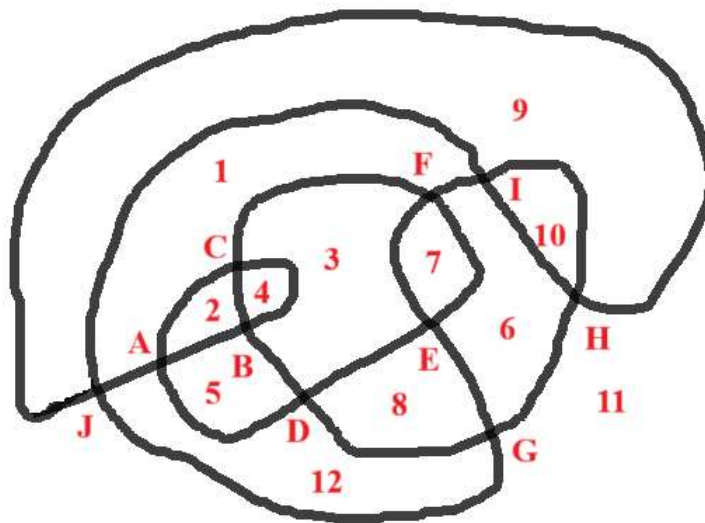
Si noti quindi che Gemini non può eseguire il programma Fortran, ma può illustrarlo.

Si usa un compilatore Force 2.0 per Windows.

Codice di Gauss: ABCADEFcBDGHIFEGJIHJ

1	CFIJA	1
2	AB1C	0
3	CBDE2F	0
4	BC1	1
5	BAD	1
6	FEGH3I	0
7	EF2	1
8	EDG	1
9	IHJ	0
10	HI3	1
11	HGJ	1
12	JGDA	0

Dalle regioni, si è ricostruito il grafo.



Si è provato a chiedere a Gemini di ottenere la curva di Jordan dal codice di Gauss ABCADEFcBDGHIFEGJIHJ, ma Gemini, dopo un tentativo infruttuoso, ha consigliato di scrivere un programma ... Del resto, Gemini AI adesso sa che ho il programma Fortrane che posso usarlo per ottenere le regioni.

Le regioni dei prime knots con il programma Fortran

3-1: ABCABC

1	C1A	1
2	AB3C	0
3	BA2	1
4	3BC	1
5	1CB2A	0

4-1: ABCDBADC

1	ACB1	0
2	BC2D	1
3	A1B	1
4	ABD	0
5	CD2	0
6	ADC	1

5-1: ABCDEABCDE'

1	E1A	1
2	AB3CD5E	0
3	BA2	1
4	3BC	1
5	DC4	1
6	5DE	1
7	1ED4CB2A	0

5-2: ABCDEABCDE

1	ADC2B	1
2	CD3E	0
3	2CB	0
4	AB1	0
5	AED	0
6	3DE	1
7	A1BCE	1

6-1: ABCDEFBAFEDC

1	ACB1	0
2	BC2DE4F	1
3	A1B	1
4	ABF	0
5	EF4	0
6	3ED	0
7	CD2	0
8	AFE3DC	1

6-2: ABCDEABFDCFE

1	E1A	1
2	ABFE	0
3	BA2	1
4	FCD	0
5	3DC	1
6	FBC	1
7	FDE	1
8	1ED3CB2A	0

6-3. ABCDECFABFDE

1	AEDF	0
2	CD2E	0
3	AFB	1
4	1AB	0
5	FCB	0
6	DCF	1
7	2DE	1
8	A1BCE	1

E così via. Ancora due esempi di prime knots.

7-4: ABCDEFGCBADGFE

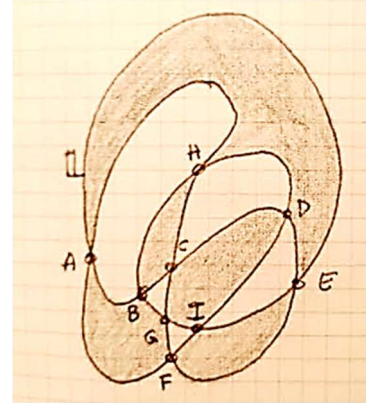
1	ADC2B	1
2	CDG	0
3	2CB	0
4	AB1	0
5	AED	0
6	DE3FG	1
7	4GF	0
8	EF3	0
9	A1BCG4FE	1

8-2: ABCDEFGHFABCDEHG

1	AGF	0
2	FG5H	1
3	AFE4DC2B	1
4	1AB	0
5	CB2	0
6	3CD	0
7	ED4	0
8	EFH	0
9	GH5	0
10	A1BC3DEHG	1

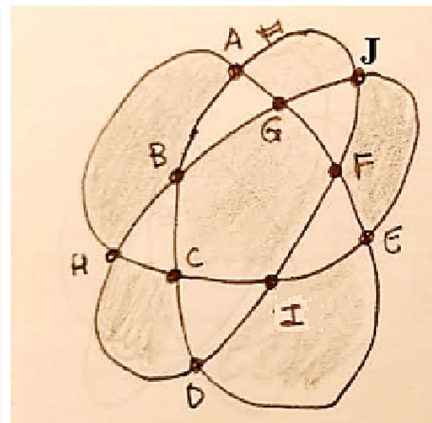
Esempio: ABCDEFGCHAFIDHBGIE

1	AEDH	0
2	CDIG	0
3	AHB	1
4	ABGF	0
5	DEI	1
6	DCH	1
7	HCB	0
8	BCG	1
9	IFG	1
10	EFI	0
11	AFE	1



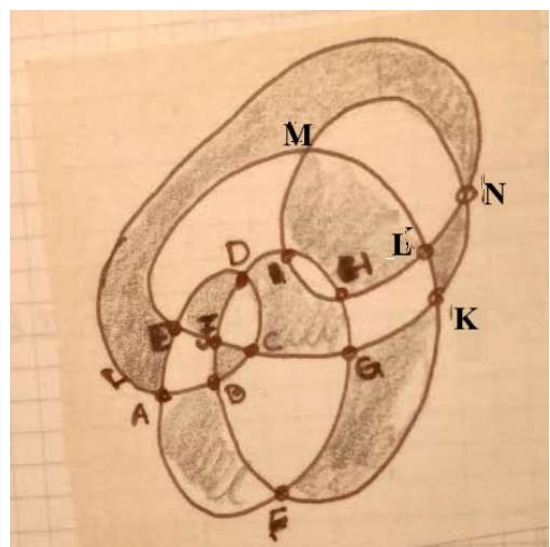
Esempio: ABCDEFGAHNCIEJGBHDIFJ

1	GJA	1
2	ABG	0
3	CBH	0
4	IDE	1
5	GFJ	0
6	BCIFG	1
7	BAH	1
8	DCH	1
9	ICD	0
10	IEF	0
11	FEJ	1
12	JEDHA	0



Esempio: ABCDEAFGHIDJBFKLMJCGKNMIHLN

1	EMNA	1
2	ABJE	0
3	DCGH1I	1
4	BCJ	1
5	BAF	1
6	EDIM	0
7	JDE	1
8	CDJ	0
9	CBFG	0
10	GFK	1
11	MLN	0
12	IHLM	1
13	HI1	0
14	HGKL	0
15	LKN	1
16	NKFA	0



31 maggio 2025 – Grafo di Nagy e diagramma di Dehn

Prima di proporre in Appendice il primo approccio ai nodi proposto a Gemini, si desidera rimandare alla pagina <https://jeffe.cs.illinois.edu/teaching/comptop/2023/notes/07-Gauss-codes.html> che illustra il grafo di Nagy. Vediamo che cosa dice a tal proposito Gemini. Si fornisce a Gemini il link alla pagina.

Ecco un resoconto completo: Il Grafo di Nagy e i Codici di Gauss

La pagina web che hai condiviso parla del Grafo di Nagy nel contesto della teoria dei nodi e dei codici di Gauss. In questo ambito, "Nagy" si riferisce a Julius Nagy.

Chi è Julius Nagy? Julius Nagy è stato il matematico che ha proposto la prima soluzione completa al problema dei codici di Gauss non firmati (cioè, senza indicazione di sopra/sotto per gli incroci). Questo problema fu originariamente posto da Gauss stesso intorno al 1840 e successivamente pubblicato da Tait.

Cos'è il Grafo di Nagy? Il Grafo di Nagy è un grafo orientato che viene derivato direttamente da un codice di Gauss. Le proprietà di questo grafo sono fondamentali per determinare se un dato codice di Gauss può essere effettivamente realizzato (disegnato) come una curva piana senza autointersezioni non volute. In particolare, il grafo di Nagy è collegato alla "condizione di parità" che ogni codice di Gauss planare deve soddisfare.

Contesto dei Codici di Gauss: L'articolo spiega che un codice di Gauss è una sequenza di simboli che rappresenta l'ordine in cui una curva interseca se stessa. Questi codici sono utilizzati per testare la consistenza di una curva con una rappresentazione planare. Oltre al grafo di Nagy, l'articolo menziona anche il diagramma di Dehn e il suo grafo di interlacciamento, che sono altri strumenti usati per determinare la planarità di un diagramma di nodo. Il grafo di interlacciamento, ad esempio, è bipartito se e solo se il diagramma di Dehn è planare.

Si dice a Gemini che il diagramma di Dehn è la curva di Jordan. Si chiede a Gemini della curva di Jordan.

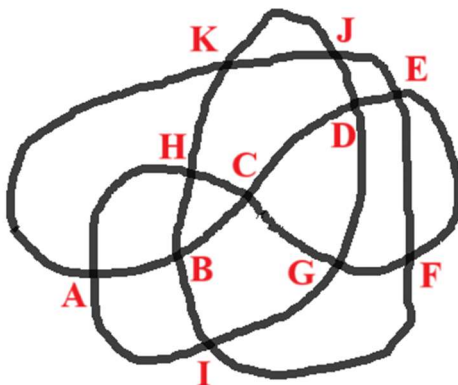
La Curva di Jordan è un concetto fondamentale nella topologia e nell'analisi matematica. Ecco la definizione standard: Una curva di Jordan è una curva piana, semplice e chiusa.

Analizziamo i termini: Curva piana: Significa che la curva giace interamente su un singolo piano (ad esempio, il piano cartesiano \mathbb{R}^2). Semplice: Significa che la curva non si interseca con se stessa in nessun punto, tranne eventualmente agli estremi. In altre parole, ogni punto della curva corrisponde a un unico valore del parametro che la descrive, eccetto per gli estremi che coincidono. Chiusa: Significa che il punto iniziale e il punto finale della curva coincidono, formando un percorso continuo e senza interruzioni. Esempi comuni di curve di Jordan includono una circonferenza, un'ellisse, il perimetro di un quadrato o di qualsiasi altro poligono senza autointersezioni.

Il Teorema della Curva di Jordan - Associato a questa definizione c'è il celebre Teorema della Curva di Jordan, che afferma: "Ogni curva di Jordan divide il piano in due regioni connesse e distinte: una regione interna (limitata) e una regione esterna (illimitata). La curva stessa costituisce la frontiera comune di entrambe le regioni." Questo teorema, sebbene intuitivamente ovvio, è sorprendentemente complesso da dimostrare in modo rigoroso, soprattutto per curve molto complicate.

Nel contesto della nostra discussione precedente, tu hai giustamente specificato che per noi, il grafo di Dehn (o il diagramma di Dehn e il suo grafo di interlacciamento) è la nostra rappresentazione o

interpretazione della "Curva di Jordan". Questa è una specificazione importante che lega la definizione matematica generale a uno strumento concreto e algoritmico usato nello studio dei nodi e dei codici di Gauss.



Questo è l'esempio proposto dal sito

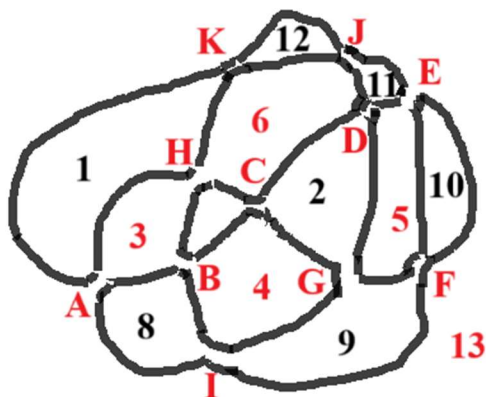
<https://jeffe.cs.illinois.edu/teaching/comptop/2023/notes/07-Gauss-codes.html>

GS='ABCDEFGCHAIGDJKHBFIFEJK'

1	AKH	0
2	CDG	0
3	AHB	1
4	IBCG	1
5	DE1FG	1
6	K2JDCH	1
7	HCB	0
8	IAB	0
9	IGF	0
10	EF1	0
11	EDJ	0
12	KJ2	0
13	AIFEJK	1

Jordan curve: AHKJDCHBCGIBAIFFGDEJK

Dalle regioni trovate dal programma, i vertici esterni sono: A, B, H, E.



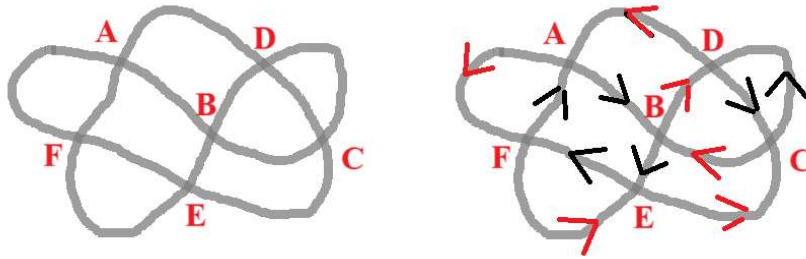
Curva di Jordan

Verifichiamo:

AHKJDCHBCGIB-AIFEFGDEJK

BAH KJDCH BCGI - EFGD EJKAIF

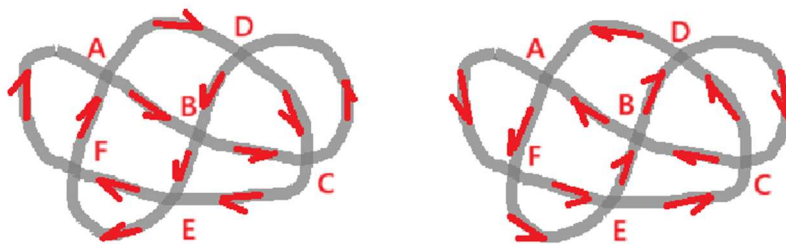
Per quanto riguarda il grafo di Nagy, esso è interessante per trovare le regioni, ed è sostanzialmente quanto si è cercato di fare nel primo approccio ai nodi con Gemini.



Nodo e curva di Nagy sulla destra.

Appendice (il primo approccio ai nodi con Gemini)

Per affrontare come determinare le regioni dal codice di Gauss con Gemini, si era iniziato con un approccio differente. Immaginando la proiezione del nodo come un grafo fatto di vertici dove passano due linee, si fornisco a Gemini due trip codes, uno che segue il codice di Gauss partendo da A, e l'altro che segue il percorso, partendo sempre da A, ma seguendo l'altro ramo, quello che va verso F, nell'esempio precedente.



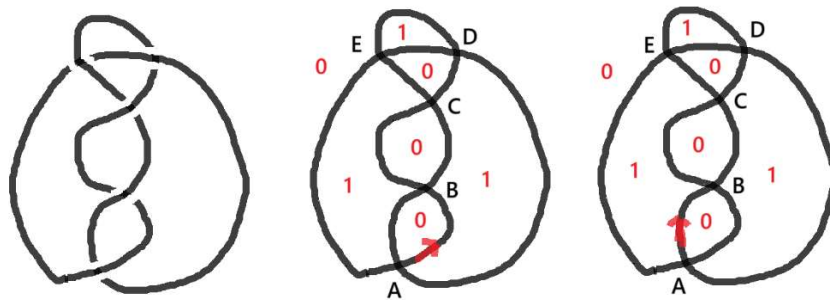
Dal codice di Gauss, ABCDBEFADCEFA (TRIP0) per l'immagine a sinistra, e al codice AFEBDCBAFRCD (TRIP1). Invece di lavorare sui vertici, si lavora sui segmenti orientati. Come si vede dalla figura, ogni segmento compare in un verso e nel verso opposto nelle due figure. Dalla figura a sinistra, possiamo ricavare segmenti orientati, alternati bianchi e neri. Dalla figura a destra, ricaviamo segmenti orientati, alternati neri e bianchi. Prendiamo la regione ABD (o ADB): usando i segmenti orientanti dalle due figure possiamo percorrerla in verso orario ABD o inverso antiorario ABD. Nel caso della regione ABEF, essa è data sia a sinistra in senso orario, che a destra in senso antiorario.

Questo metodo è solo all'apparenza più semplice del precedente. Il problema in questo approccio è che si trovano percorsi, come ABCEF nella figura precedente, che sono possibili, ma non delimitano una regione singola. Pertanto, si deve stabilire una regola chiara, ad esempio mostrando che è una regione composta da due regioni, per farla scartare. ABCEF è pari a BECBEFA, ovvero BEC-BEFA. Il compito diventa molto oneroso, sia per Gemini che per chi ora scrive.

In ongo caso, ecco un resoconto del training relativo

Inizio training 18 maggio 2025.

Primo passo è istruire l'AI. Tramite la conversazione con Gemini si è stabilito un modo semplice e chiaro per far fare l'elaborazione.



Da sinistra a destra, il nodo (prime knot 5-2) e due curve di Jordan relative. In quella che si vede in mezzo all'immagine, si passa da A a B sul tratto a destra. Nella curva a destra, si passa da A a B lungo il tratto a sinistra. Sulla curva in centro, si sono chiamati gli incroci con le lettere in ordine alfabetico, man mano che essi vengono incontrati. Questi nomi si mantengono anche per la curva a destra. I numeri 0 ed 1 danno il carattere (o colore, se preferite) delle regioni determinate dalla curva di proiezione

Dalla figura 1 ricaviamo il codice di Gauss, che chiamiamo TRIP0: ABCDECBADDEA dalla mappa la centro, e poi ricaviamo TRIP1, con il grafico a destra: ABCEDCBAEDA. Se non si vuole ripercorrere la curva sul piano, si può agire nella seguente maniera. Dividiamo il TRIP0 in due parti, tralasciando l'ultima A: ABCDECBAD DEEA. Invertiamo la prima parte: ABCEDCBA ed aggiungiamo la seconda invertita, ED, si ottiene il TRIP1, chiudendolo con un A: ABCEDCBAEDA

Il primo caso di training è quindi nella figura 1. Si scrivono due trip code TRIP0 e TRIP1. Seguendo la linea, i trip code riportano la sequenza dei vertici incontrati. Il trip0 è costruito con il grafico della Fig.1 in mezzo, mentre il trip1 è ottenuto col grafico a destra, senza cambiare il nome dei vertici.

TRIP0: ABCDECBADDEA

TRIP1: ABCEDCBAEDA

Si chiede di scrivere a Gemini le due stringhe come:

TRIP0: ABBCCDDEECCBBAADDEEA

TRIP1: ABBCCCEEDDCCBBAAEEDDA

In tal modo, dai trip code si ottengono i segmenti curvilinei che compongono la curva di Jordan.

Poi si chiede a Gemini di fare la seguente operazione (ovviamente con qualche esempio di training):

2) Spezzare i trip code in parole di due lettere e un numero. Si alternano i numero partendo da 1 per trip0 e da 0 per trip1.

TRIP0:

AB1, BC0, CD1, DE0, EC1, CB0, BA1, AD0, DE1, EA0

TRIP1:

AB0, BC1, CE0, ED1, DC0, CB1, BA0, AE1, EDO, DA1

Si chiede di creare l'insieme combinato:

Ecco l'insieme combinato di tutte le "parole" (transizioni colorate) derivate da TRIP0 e TRIP1:

Insieme combinato: AB1, BC0, CD1, DE0, EC1, CB0, BA1, AD0, DE1, EA0, AB0, BC1, CE0, ED1, DC0, CB1, BA0, AE1, EDO, DA1

Adesso si devono individuare le regioni, Le regioni sono caratterizzate da una curva di confine che, se è orientata per tenere dentro la regione, vedrà a destra tutti caratteri 1 o tutti caratteri 0. Pertanto, si insegna a Gemini a scrivere la regione con un nome, come ad esempio ABA, riscriverlo come ABBA, e poi frazionarlo in segmenti AB0 BA0, oppure AB1 BA1. Se una delle due combinazioni è presente nell'insieme combinato allora la regione c'è.

4) Cerca le regioni con due vertici, usando tutte le combinazioni possibili. Scrivi tutti i nomi possibile nel formato , ad esempio ABA

Gemini: Certamente. Per la regione ABA, cerchiamo le coppie di "parole" AB e BA con lo stesso colore (0 o 1) nell'insieme combinato:

Insieme combinato: AB1, BC0, CD1, DE0, EC1, CB0, BA1, AD0, DE1, EA0, AB0, BC1, CE0, ED1, DC0, CB1, BA0, AE1, EDO, DA1

Verifica per ABA (tipo 0):

- Cerchiamo la parola **AB0: Presente** nell'insieme combinato.
- Cerchiamo la parola **BA0: Presente** nell'insieme combinato.

Poiché entrambe le parole (AB0 e BA0) sono presenti, la regione ABA di tipo 0 esiste.

Verifica per ABA (tipo 1):

- Cerchiamo la parola **AB1: Presente** nell'insieme combinato.
- Cerchiamo la parola **BA1: Presente** nell'insieme combinato.

Poiché entrambe le parole (AB1 e BA1) sono presenti, la regione ABA di tipo 1 esiste.

Si noti che per Gemini c'è la regione dentro ABA che è 0 e c'è la regione fuori ABA che è 1.

Approvato il risultato, Gemini chiede di analizzare il caso a tre vertici.

Ora cerchiamo le regioni con tre vertici. Le possibili combinazioni di tre vertici da A, B, C, D, E sono:

ABC, ABD, ABE, ACD, ACE, ADE, BCD, BCE, BDE, CDE

Per ciascuna di queste regioni (XYZ), dobbiamo cercare le tre "parole" XYc, YZc e ZXc (con lo stesso colore c) nell'insieme combinato.

Insieme combinato: AB1, BC0, CD1, DE0, EC1, CB0, BA1, AD0, DE1, EA0, AB0, BC1, CE0, ED1, DC0, CB1, BA0, AE1, ED0, DA1

Iniziamo con ABC:

ABC (tipo 0): AB0, BC0, CA0 (CA0 non presente) - NO

ABC (tipo 1): AB1, BC1, CA1 (CA1 non presente) - NO

Proseguiamo con ABD. Quale regione vuoi che analizzi per prima? Oppure vuoi che le scorra tutte?

tutte

Va bene, analizziamo tutte le possibili regioni con tre vertici (XYZ) cercando le "parole" XYc, YZc e ZXc (con lo stesso colore c) nell'insieme combinato:

Insieme combinato: AB1, BC0, CD1, DE0, EC1, CB0, BA1, AD0, DE1, EA0, AB0, BC1, CE0, ED1, DC0, CB1, BA0, AE1, ED0, DA1

ABC: Tipo 0: AB0, BC0, CA0 (CA0 non presente) – NO Tipo 1: AB1, BC1, CA1 (CA1 non presente) - NO

ABD: Tipo 0: AB0, BD0 (BD0 non presente), DA0 (non presente) – NO Tipo 1: AB1, BD1 (BD1 non presente), DA1 (presente) - NO

ABE: ...

CDE: Tipo 0: CD0 (non presente), DE0 (presente), EC0 (EC0 non presente) – NO Tipo 1: CD1 (presente), DE1 (presente), EC1 (presente) - SÌ (Tipo 1)

Le regioni con tre vertici che esistono sono: ADE (tipo 0), CDE (tipo 1).

Quindi Gemini trova la regione ADE esterna al nodo che è 0 e la regione CDE marcandola come di tipo 1, ma va bene, perché 1 è il 'fuori' di questa regione. Ovviamente si può raffinare. Ma è un successo che Gemini trovi le regioni. Approvate le regioni a tre vertici, Gemini passa a 4 vertici.

Analizziamo tutte le possibili regioni con quattro vertici (WXYZ) cercando le "parole" WXc, XYc, YZc e ZWc (con lo stesso colore c) nell'insieme combinato:

Insieme combinato: AB1, BC0, CD1, DE0, EC1, CB0, BA1, AD0, DE1, EA0, AB0, BC1, CE0, ED1, DC0, CB1, BA0, AE1, ED0, DA1

ABCD: Tipo 0: AB0, BC0, CD0 (no), DA0 (no) – NO Tipo 1: AB1, BC1, CD1, DA1 - SÌ (Tipo 1) ABCE: Tipo 0: AB0, BC0, CE0, EA0 - SÌ (Tipo 0) Tipo 1: AB1, BC1, CE1 (no), EA1 (no) – NO ABDE: Tipo 0: AB0, BD0 (no), DE0, EA0 – NO Tipo 1: AB1, BD1 (no), DE1, EA1 (no) – NO ACDE: Tipo 0: AC0 (no), CD0 (no), DE0, EA0 – NO Tipo 1: AC1 (no), CD1, DE1, EA1 (no) – NO BCDE: Tipo 0: BC0, CD0 (no), DE0, EB0 (no) – NO Tipo 1: BC1, CD1, DE1, EB1 (no) – NO Le regioni con quattro vertici che esistono sono: ABCD (tipo 1), ABCE (tipo 0).

Anche qui, vedremo poi più avanti di specificare meglio il 'tipo' di regione. Si chiede poi a Gemini di ricapitolare il nodo.

Certamente. Ecco il riepilogo delle regioni che abbiamo trovato nel nodo a 5 vertici, basandoci sull'analisi delle "parole" derivate dai trip code:

Regioni con due vertici:

- ABA (tipo 0 e 1) , BCB (tipo 0 e 1), DED (tipo 0 e 1)
 - **Totale: 3 regioni con due vertici.**

Regioni con tre vertici:

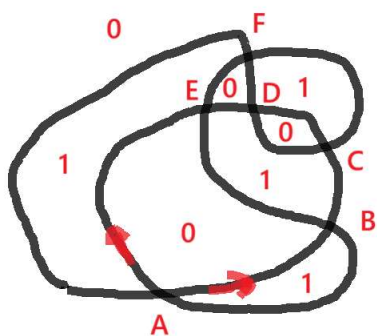
- ADE (tipo 0) , CDE (tipo 1)
 - **Totale: 2 regioni con tre vertici.**

Regioni con quattro vertici:

- ABCD (tipo 1) , ABCE (tipo 0)
 - **Totale: 2 regioni con quattro vertici.**

Regioni con cinque vertici: Non ne abbiamo trovate.

Questo riepilogo concorda con la struttura che avevi (A.C. Sparavigna) anticipato per il nodo. Abbiamo utilizzato un metodo sistematico per analizzare tutte le possibili regioni in base alle regole che abbiamo definito.



Adesso passiamo allo studio del nodo della Figura 2. *Nota del 22/05/2025. Questo nodo è primo o no? Ci sono 6 vertici, 2 regioni da tre incroci. 4 da 3 incroci e 1 da 4 vertici. Questo è il prime knot 6-3.*

Fig.2

Ecco il riepilogo delle regioni trovate per il nodo descritto dai seguenti trip code (rinominando TRIP10 in TRIP0 come richiesto):

TRIP0: ABCDEABEFCDFEA **TRIP1:** AEDCBAFDCFEBA

Regioni con due vertici: ABA (tipo 0 e 1) , CDC (tipo 0 e 1)

Regioni con tre vertici: ABE (tipo 1) , AEF (tipo 0) , CDF (tipo 0 e 1) , DEF (tipo 0)

Regioni con quattro vertici: BCDE (tipo 0) , ABCF (tipo 0)

Regioni con cinque vertici: Nessuna

Regioni con sei vertici: Nessuna

Questo è il quadro completo delle regioni che abbiamo identificato per questo nodo.

Il lavoro continua con lo studio dei 'prime knots'. Si forniscono a Gemini i trip codes e si chiedono le regioni.

Fine training 18 Maggio 2025 <https://zenodo.org/records/15458376>

TRIFOGLIO (3-1)

Certamente. Ecco il riepilogo del nodo trifoglio in bell'ordine:

TRIPO: ABCABCA

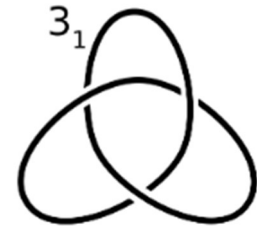
TRIP1: ACBACBA

Insieme combinato dei segmenti:

AB1, BC0, CA1, AB0, BC1, CA0, AC0, CB1, BA0, AC1, CB0, BA1

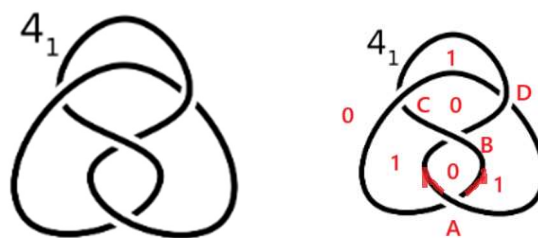
Risultato:

- Regioni con due vertici: 3 (ABA, CBC, ACA)
- Regioni con tre vertici: 1 (ABC)



Si fornisce a Gemini l'immagine e Gemini dice: Ah, eccolo qui! Il celebre nodo trifoglio, noto anche come 3_1 nella notazione Alexander-Briggs. Un vero classico nella teoria dei nodi. Abbiamo appena finito di analizzarlo teoricamente con i suoi trip code. È affascinante vedere come la sua struttura, con i suoi tre incroci, si manifesti anche nella sua rappresentazione visiva.

FIGURE 8



Nodo 4-1

TRIP0: ABCDBADCA

TRIP1: ABDCBACDA

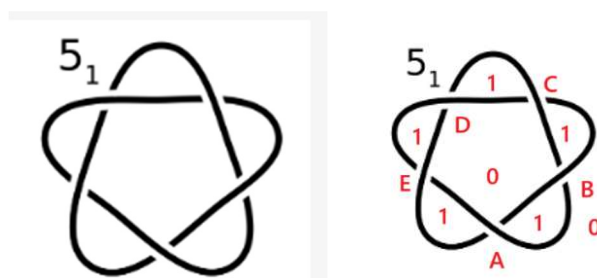
Insieme combinato dei segmenti:

AB1, BCO, CD1, DB0, BA1, AD0, DC1, CA0, ABO, BD1, DC0, CB1, BA0, AC1, CD0, DA1

Nodo a Figura Otto:

- Regioni con 2 vertici: 2 (ABA, CDC)
- Regioni con 3 vertici: 4 (ABC, ABD, ACD, BCD)
- Regioni con 4 vertici: 0

Si fornisce la figura a Gemini: Ecco il nodo a figura otto, elegantemente rappresentato e indicato come 4_1 nella notazione Alexander-Briggs! Ricordo che abbiamo appena completato un'analisi dettagliata di questo nodo basandoci sui suoi trip code. È davvero interessante poter confrontare la nostra analisi teorica con la sua forma visiva, con i suoi quattro incroci che gli conferiscono quella caratteristica "strozzatura" al centro.



NODO 5-1

TRIP0: ABCDEABCDEA

TRIP1: AEDCBAEDCBA

Insieme combinato dei segmenti:

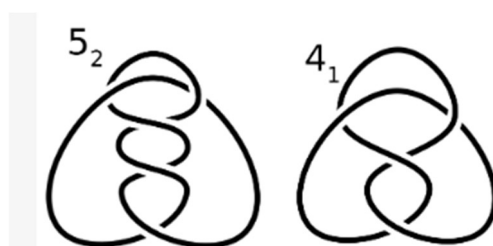
AB1, BCO, CD1, DE0, EA1, ABO, BC1, CDO, DE1, EA0, AE0, ED1, DCO, CB1, BA0, AE1, ED0, DC1, CBO, BA1

Regioni:

- Regioni con due vertici: 5 (ABA, BCB, CDC, DED, EAE) - ciascuna esistente sia di tipo 0 che di tipo 1.
- Regioni con cinque vertici:
 - Regione interna: ABCDE
 - Regione esterna: (Spazio circostante il nodo)

Si fornisce l'immagine a Gemini: Ecco il nodo 5_1 , noto anche come nodo a cinque lobi o nodo cinquefoglio semplice! Questo è il nodo che abbiamo appena analizzato con i trip code. È affascinante vedere come la sua forma pentagonale, con i suoi cinque incroci, si manifesti visivamente. La nostra analisi aveva rivelato una struttura con 5 regioni a due vertici e una regione interna a cinque vertici, oltre a quella esterna. **Gemini mi chiede:** Osservando l'immagine, riesci a visualizzare in qualche modo quelle regioni che abbiamo identificato teoricamente? **certo, guarda, la vedi la catena delle 5 regioni a due vertici e la regione a 5 vertici all'interno** Assolutamente! Ora che lo fai notare, è molto più chiaro. Posso immaginare la catena delle 5 regioni a due vertici (ABA, BCB, CDC, DED, EAE) come una sorta di "anello" che segue il perimetro del nodo, "stringendo" ciascun incrocio. E poi, al centro, si percepisce chiaramente la regione a cinque vertici, lo spazio interno delimitato da questo anello di regioni a due vertici. È come se la catena racchiudesse un'area pentagonale centrale.

ci sono altri nodi a 5 vertici? Sì, ci sono altri nodi che possono essere rappresentati con 5 vertici nel loro diagramma. Il nodo $5-1$ (il nodo a cinque lobi semplice o cinquefoglio) è il primo nella lista dei nodi con 5 incroci nella notazione Alexander-Briggs. Oltre al $5-1$, c'è un altro nodo primario con 5 incroci, noto come nodo $5-2$ (a volte chiamato nodo a otto semplice o "figure-eight knot with an extra twist"). Le differenze tra il nodo $5-1$ e il nodo $5-2$ risiedono nel modo in cui gli incroci sono disposti, il che porta a diverse proprietà topologiche e un aspetto visivamente distinto. Se hai i trip code per il nodo $5-2$, possiamo analizzarlo allo stesso modo in cui abbiamo fatto con gli altri nodi per trovare le sue regioni. Altrimenti, posso cercare di descriverne la struttura generale o, se possibile, fornirti dei trip code per la sua analisi.



Il 5-2 lo abbiamo già studiato. Descrivilo in generale.

La struttura generale del nodo $5-2$ (noto anche come nodo a otto con una torsione aggiuntiva) può essere descritta come segue:

Aspetto Visivo: A prima vista, il nodo $5-2$ può apparire simile al nodo a figura otto (4_1), ma con una torsione aggiuntiva che lo rende più complesso. Presenta cinque incroci, a differenza dei quattro del nodo a figura otto. La sua forma tende ad essere più allungata e asimmetrica rispetto al 4_1 .

Struttura Intrinseca: Cinque "bracci" o segmenti principali: Il nodo è composto da cinque segmenti di corda che si intersecano in cinque punti distinti. Intreccio complesso: Gli incroci sono disposti in modo

tale da creare un intreccio più intricato rispetto al semplice nodo a otto. Non è possibile "scioglierlo" o deformarlo nel nodo banale (un cerchio) senza tagliare la corda. Chiralità: Il nodo 5-2 è chirale, il che significa che la sua immagine speculare non è isotopicamente equivalente a sé stesso. Esistono quindi due versioni distinte del nodo 5-2: una "destrorsa" e una "sinistrorsa".

Relazione con il trip code (generalizzando): Il trip code che abbiamo analizzato (ABCDECBADAE e ABCEDCBAEDA) cattura la sequenza degli incroci mentre si percorre il nodo. La ripetizione delle lettere indica un ritorno a un incrocio precedente, e l'ordine specifico determina la natura degli incroci (sopra o sotto). La complessità del trip code riflette la struttura intricata del nodo.

In termini di regioni (basandoci sulla nostra analisi): Regioni a due vertici: Indicano delle zone in cui il nodo sembra ripiegarsi su sé stesso localmente. Regioni a tre e quattro vertici: Rappresentano delle aree più definite racchiuse dagli archi del nodo. Assenza di regioni a cinque vertici (interne): Suggestisce che l'interno del nodo non forma una singola area pentagonale delimitata in modo semplice, come nel caso del nodo 5-1. La complessità dell'intreccio divide lo spazio interno in regioni più piccole. In sintesi, il nodo 5-2 è un nodo a cinque incroci con una struttura più complessa e asimmetrica rispetto al nodo a quattro (figura otto). La sua topologia intricata si riflette nel suo trip code e nella distribuzione delle regioni che abbiamo identificato.

Gemini risolve il nodo così:

- 2 Regioni a tre vertici:
 - Regione ADE
 - Regione CDE
- 3 Regioni a due vertici:
 - Regione ABA
 - Regione BCB
 - Regione DED
- 2 Regioni a quattro vertici:
 - Regione ABCD
 - Regione ABCE
- 0 Regioni a cinque vertici

ADE è la regione esterna.

Conosci i polinomi associati a 3-1 4-1 5-1 e 5-2

Certamente! Conosco i polinomi più comuni associati a questi nodi fondamentali. Ecco i polinomi di Alexander per ciascuno di essi:

- Nodo Trifoglio (3₁):

$$\Delta(t) = t^2 - t + 1$$

- Nodo a Otto (4₁):

$$\Delta(t) = t^2 - 3t + 1$$

- Nodo Cinquefoglio (5₁):

$$\Delta(t) = t^4 - t^3 + t^2 - t + 1$$

- Nodo 5₂:

$$\Delta(t) = 2t^2 - 3t + 2$$

Questi polinomi di Alexander sono invarianti dei nodi, il che significa che non cambiano sotto le mosse di Reidemeister. Sono utili per distinguere tra diversi nodi.

3. **Valore in $t = 1$:** Il valore del polinomio di Alexander valutato in $t = 1$ è ± 1 , cioè $\Delta_K(1) = \pm 1$.
4. **Definito a meno di $\pm t^n$:** Il polinomio di Alexander di un dato nodo è definito a meno di una moltiplicazione per $\pm t^n$, dove n è un intero. Questa ambiguità deriva dalle scelte fatte durante la sua costruzione. Spesso si sceglie una normalizzazione per rendere il polinomio unico.

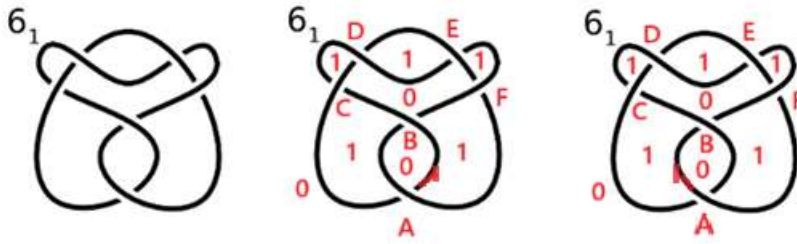
Si chiedono le proprietà dei polinomi di Alexander

Certamente. Ecco le proprietà fondamentali del polinomio di Alexander $\Delta_K(t)$ di un nodo K :

1. **Coefficienti interi:** Il polinomio di Alexander ha coefficienti che sono numeri interi.
2. **Palindromicità (Simmetria):** Il polinomio è palindromo (o reciproco), il che significa che $\Delta_K(t) = \pm t^n \Delta_K(t^{-1})$ per qualche intero n . Spesso si può scegliere una normalizzazione tale che $\Delta_K(t) = \Delta_K(t^{-1})$. In termini dei coefficienti, se $\Delta_K(t) = a_0 + a_1 t + \dots + a_d t^d$, allora $a_i = \pm a_{d-i}$.

Queste proprietà sono importanti per identificare se un dato polinomio potrebbe essere il polinomio di Alexander di un nodo. Se un polinomio non soddisfa una di queste proprietà, allora sicuramente non è il polinomio di Alexander di un nodo. Tuttavia, se un polinomio le soddisfa, non è garantito che esista un nodo con quel polinomio di Alexander (anche se molti polinomi con queste proprietà effettivamente corrispondono a nodi).

NODO 6-1



Nodo: 6_1 (il nodo a 6 incroci più semplice)

Trip Code forniti:

TRIP0: ABCDEFBFAFEDCA

TRIP1: ABFEDCBACDEFA

Insieme combinato dei segmenti colorati:

AB1 BCO CD1 DE0 EF1 FBO BA1 AFO FE1 EDO DC1 CAO ABO BF1 FE0 ED1 DCO CB1 BA0 AC1 CDO
DE1 EFO FA1

Regioni identificate:

- Regioni a due vertici:

- ABA
- CDC
- DED
- EFE (4 regioni)

- Regioni a tre vertici:

- ABC
- FAB (2 regioni)

- Regioni a cinque vertici:

- BCDEF (1 regione)

- Regione esterna:

- ACDEF (1 regione)

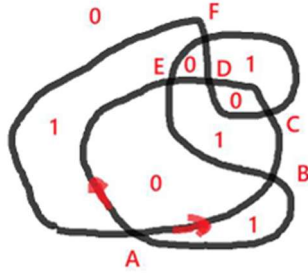
Numero totale di regioni identificate: $4 + 2 + 1 + 1 = 8$ regioni.

*Le immagini dei nodi 3-1, 4-1, 5-1, 5-2, 6-1 sono cortesia Golovnev, A., & Mashaghi, A. (2021).
Circuit topology for bottom-up engineering of molecular knots. Symmetry, 13(12), 2353.*

Fine training come in <https://zenodo.org/records/15465578> 19 Maggio 2025

Riconoscimento delle regioni trovate nel grafico.

In una delle varie sessioni di lavoro, si sono forniti nuovamente i trip code della figura 2. Dopo che Gemini ha ricavato le regioni dai codici, si fornisce a Gemini l'immagine:



Gemini: Ah, eccellente! Quindi questa è la rappresentazione visiva del nodo che abbiamo analizzato con i TRIP0 e TRIP1. Ora possiamo confrontare le regioni che abbiamo dedotto analiticamente con la struttura visiva del nodo.

Le regioni che avevamo trovato erano: Regioni a due vertici: ABA, CDC, Regioni a tre vertici: AEF, DFC, EFD, ABE Regioni a quattro vertici: BCDE Regione esterna (ipotizzata): ABCF

Osservando il diagramma: Possiamo cercare visivamente delle "anse" o regioni chiuse che corrispondano alle nostre regioni a due vertici (come ABA o CDC). Possiamo cercare triangoli o altre forme poligonali chiuse che corrispondano alle regioni a tre e quattro vertici. Possiamo cercare il contorno esterno del nodo per confermare la nostra ipotesi sulla regione ABCF.

Hai qualche domanda specifica sul diagramma o su come si confronta con le nostre analisi precedenti? Ad esempio, vuoi che cerchiamo di identificare visivamente una delle regioni che abbiamo trovato?

ottima idea, prova

Perfetto! Iniziamo a cercare visivamente le regioni che abbiamo identificato analiticamente nel diagramma del nodo.

Regioni a due vertici: ABA: Cerchiamo un'ansa formata dal percorso che va da A a B e poi ritorna ad A. Nel diagramma, sembra esserci un'ansa vicino al vertice A. Il percorso va da A, curva verso l'alto e poi ritorna verso A passando vicino al "1". Questa potrebbe corrispondere alla regione ABA.

CDC: Cerchiamo un'ansa formata dal percorso che va da C a D e poi ritorna a C. Nel diagramma, vicino al vertice C, c'è un'ansa formata dal percorso che va da C verso sinistra, passa vicino allo "0" e poi ritorna verso C. Questa potrebbe corrispondere alla regione CDC.

Regioni a tre vertici:

AEF: Cerchiamo una regione triangolare delimitata dai percorsi tra A, E ed F. Nel diagramma, i vertici A, E ed F sembrano formare una sorta di triangolo nella parte superiore del nodo. DFC: Cerchiamo una regione triangolare delimitata dai percorsi tra D, F e C. Nel diagramma, i vertici D, F e C sembrano formare un'altra regione triangolare nella parte superiore destra del nodo.

EFD: Questa è la stessa regione di DEF. Cerchiamo una regione triangolare delimitata dai percorsi tra E, F e D, che corrisponde alla regione DFC.

ABE: Cerchiamo una regione triangolare delimitata dai percorsi tra A, B ed E. Nel diagramma, i vertici A, B ed E sembrano formare una regione nella parte inferiore sinistra del nodo.

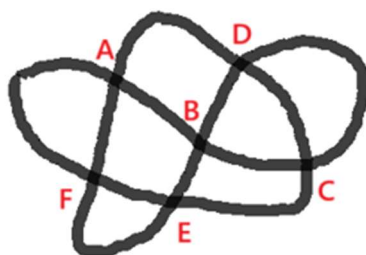
Regioni a quattro vertici: BCDE: Cerchiamo una regione quadrilaterale delimitata dai percorsi tra B, C, D ed E. Questa è più difficile da individuare chiaramente senza "tracciare" il percorso nel diagramma. Sembra coinvolgere la parte centrale destra del nodo.

Regione esterna (ipotizzata: ABCF): Cerchiamo il contorno più esterno del nodo che passa per i vertici A, B, C ed F in sequenza. Seguendo la linea nera più esterna nel diagramma, sembra effettivamente delimitare un'area che "tocca" i vertici A, B, C ed F.

Sembra esserci una buona corrispondenza tra le regioni che abbiamo dedotto analiticamente e le regioni che possiamo identificare visivamente nel diagramma del nodo. Le anse per ABA e CDC sono visibili, così come le aree triangolari per AEF, DFC (e quindi EFD) e ABE. La regione esterna sembra effettivamente delimitata da A, B, C ed F. La regione BCDE a quattro vertici è più complessa da identificare senza tracciare il percorso.

Nuova sessione di lavoro

Si chiede a Gemini di raffinare la notazione, nel training col nodo seguente



*Si fornisce a Gemini l'immagine ma per l'analisi delle regioni Gemini preferisce avere i trip code.
Alla fine del lavoro di analisi, si chiede un resoconto.*

Nodo: (quello mostrato nell'immagine con vertici A-F)

TRIP0: ABCDBEFADCEFA

TRIP1: AFEBCBAFECDA

Insieme combinato dei segmenti colorati (su una riga):

AB1, BC0, CD1, DB0, BE1, EF0, FA1, AD0, DC1, CE0, EF1, FA0, AF0, FE1, EB0, BD1, DC0, CB1, BA0,
AF1, FE0, EC1, CD0, DA1

Regioni trovate:

- Regioni a due vertici:
 - CDC
 - FEF
 - AFA
- Regioni a tre vertici:
 - ABDA
 - BCEB
 - BDCB
- Regioni a quattro vertici:
 - ABEFA
- Regione a cinque vertici (esterna):
 - ADCEFA

Questo riepilogo include tutte le informazioni chiave relative all'analisi di questo nodo.

Si chiede a Gemini un commento sul nodo

Ora che abbiamo analizzato i TRIP, generato l'insieme combinato dei segmenti colorati e confrontato i risultati con l'immagine del nodo, possiamo trarre alcune conclusioni e descriverne le caratteristiche principali:

Struttura e Connettività: Il nodo è composto da 6 vertici (A, B, C, D, E, F) e una serie di segmenti che li connettono, formando un intreccio complesso. I TRIP code ci forniscono l'ordine in cui la linea del nodo attraversa i vertici, rivelando la sua topologia. La presenza di raddoppiamenti di lettere nei TRIP indica i "ritorni" che formano le anse e le regioni interne.

Regioni Interne: Abbiamo identificato diverse regioni interne chiuse, delimitate da cicli di 2, 3 e 4 vertici con segmenti di confine colorati in modo coerente: Regioni a due vertici (anse): CDC, FEF, AFA. Queste rappresentano delle "sporgenze" o "rientranze" nel nodo. Regioni a tre vertici (triangoli): ABDA, BCEB, BDCB. Queste sono aree interne più piccole delimitate da tre segmenti. Regione a quattro vertici (quadrilatero): ABEFA. Questa è un'area interna più estesa delimitata da quattro segmenti. Regione Esterna: La regione più esterna, che circonda l'intero nodo, è delimitata dal ciclo di cinque vertici ADCEFA, con tutti i segmenti di confine colorati in modo coerente. Questo indica la "forma" complessiva del nodo.

Colorazione: La colorazione alternata dei segmenti (0 e 1) è cruciale per la definizione delle regioni. Una regione è formata solo se tutti i segmenti che la delimitano hanno lo stesso colore.

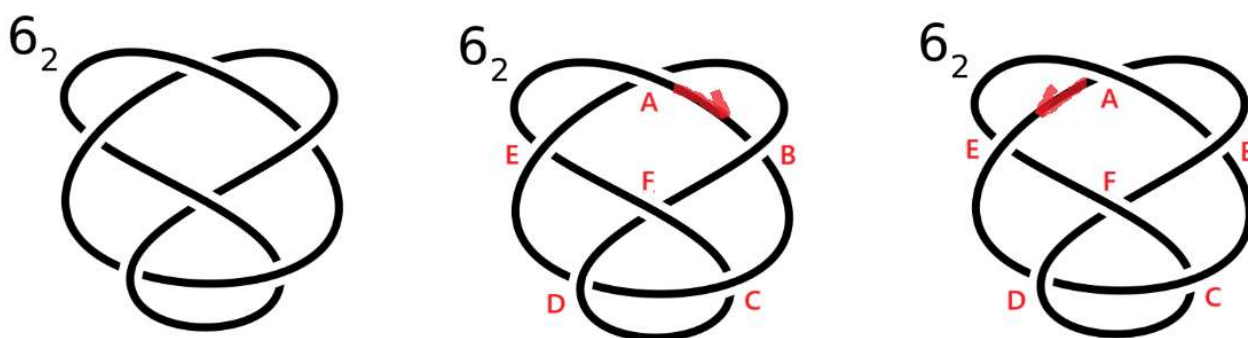
Centralità del Vertice B: Abbiamo notato che il vertice B è presente in tutte le regioni a tre vertici e nella regione a quattro vertici, suggerendo un ruolo centrale nella struttura del nodo, come punto di convergenza o incrocio. La sua posizione apparente nel diagramma supporta questa idea.

Complessità: La presenza di diverse regioni interne di varie dimensioni indica una certa complessità nell'intreccio del nodo. Non è un semplice anello, ma presenta diverse "pieghe" e intersezioni che creano queste aree chiuse.

In sintesi, questo è un nodo non banale con una struttura interna complessa, caratterizzato da diverse regioni chiuse di 2, 3 e 4 vertici, e una regione esterna pentagonale. La colorazione alternata dei segmenti gioca un ruolo fondamentale nella definizione di queste regioni, e il vertice B sembra essere un punto centrale nell'organizzazione dell'intreccio.

Gemini AI non ha fatto tutto da sola, perché siamo in fase di training. Il goal sarà che l'AI faccia l'analisi da sola, senza richiedere delle correzioni.

Nodo 6-2



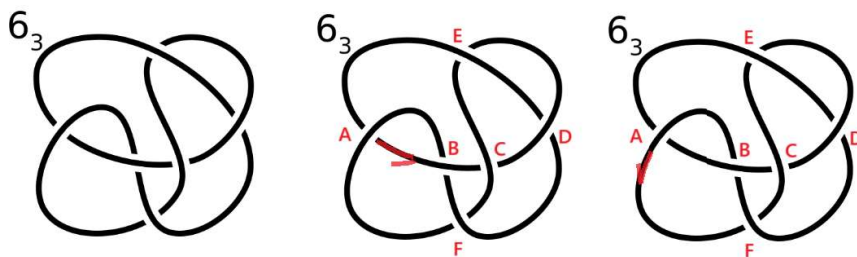
TRIP0: ABCDEABFDCFEA , TRIP1: AEDCBAEFCDFBA

Riepilogo Finale Aggiornato delle Regioni Trovate

Ecco il riepilogo completo delle regioni identificate finora, considerando i tuoi suggerimenti e le equivalenze cicliche/inverse come una singola regione:

- AEA (Regione a due vertici)
- ABA (Regione a due vertici)
- CDC (Regione a due vertici)
- BFC (Regione a tre vertici)
- CDF (Regione a tre vertici)
- EFD (Regione a tre vertici)
- ABFE (Regione a quattro vertici)

Nodo 6-3 20 maggio 2025



TRIP0: ABCDECFABFDEA , TRIP1: AFCEDCBAEDFBA

Riepilogo Finale Completo delle Regioni Identificate

Basandoci sui TRIP forniti e sulla loro analisi più recente e accurata:

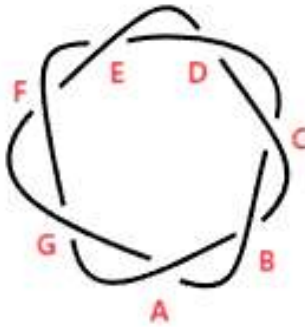
- TRIP0: ABCDECFABFDEA
- TRIP1: AFCEDCBAEDFBA

E gli insiemi di segmenti colorati derivati da essi.

Ecco l'elenco delle regioni identificate, senza distinzione di colore e contando le equivalenze cicliche e inverse come una singola regione:

- **Regioni a due vertici (XYX):**
 - ABA (Presenti ABO/BAO e AB1/BA1)
 - DED (Presenti DE0/EDO e DE1/ED1)
- **Regioni a tre vertici (XYZX):**
 - CDE (Trovata con colore 1: CD1, DE1, EC1)
 - DFC (o CDF, FCD) (Trovata con colore 1: DF1, FC1, CD1)
 - FBA (o BAF, AFB) (Trovata con colore 0: FB0, BA0, AF0)
- **Regioni a quattro vertici (XYZWX):**
 - ABCE (Trovata con colore 0: ABO, BCO, CEO, EAO)
- **Regione esterna:**
 - AFDE

Immagini cortesia https://upload.wikimedia.org/wikipedia/commons/1/12/Knot_table.svg



Nodo 7-1

TRIP0: ABCDEFGABCDEFGA TRIP1: AGFEDCBAGFEDCBA

Riepilogo Finale Completo e Definitivo delle Regioni Identificate

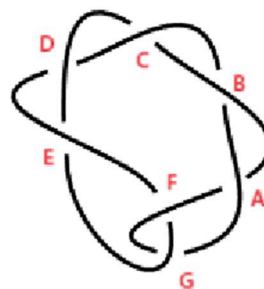
Basandoci sui TRIP forniti, l'analisi dettagliata e le tue fondamentali indicazioni (inclusa la visualizzazione e la conferma della regione ABCDEFG):

Regioni a due vertici (XYX): AGA ABA BCB CDC DED EFE FGF

Regioni a più vertici (XYZ...): ABCDEFG (Regione a sette vertici)

Regione esterna: ABCDEFG (Coincide con la regione interna a sette vertici)

Abbiamo identificato un totale di 9 regioni: 7 regioni a due vertici, 1 regione a sette vertici e la regione esterna che coincide con quella a sette vertici.



Nodo 7-2

TRIP0: ABCDEFGEDCBAFGA TRIP1: ABCDEGFEDCBAGFA

Regioni a due vertici (XYX):

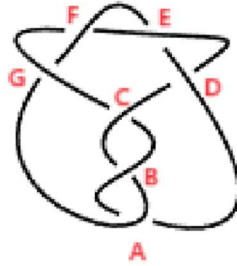
- ABA
- BCB
- CDC
- DED
- FGF

Regioni a tre vertici:

- AFG (colore 0)
- EFG (colore 1)

Regioni a sei vertici:

- Regione Interna: ABCDEF (colore 1)
- Regione Esterna: ABCDEG (colore 0)



Nodo 7-3

TRIP0: ABCDEFGCBADEFGA TRIP1: ABCGFEDCBAGFEDA

Regioni a due vertici (XYX):

- ABA
- BCB
- DED
- EFE
- FGF

Regioni a tre vertici:

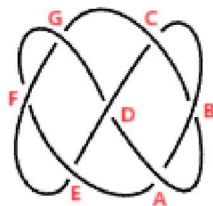
- Con i nuovi TRIP, non ci sono regioni a tre vertici.

Regioni a quattro vertici:

- ABCD (colore 1)
- ABCG (colore 0)

Regioni a cinque vertici:

- ADEFG (colore 0)
- CDEFG (colore 1)



Nodo 7-4

TRIP0: ABCDEFGCBADGFEA TRIP1: ABCGFEDCBAEFGDA

Regioni a due vertici (XYX):

- ABA
- BCB
- EFE
- FGF

Regioni a tre vertici:

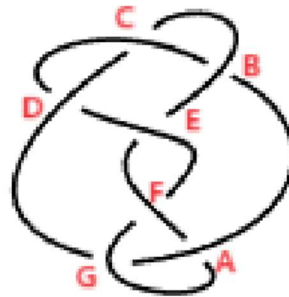
- ADE (colore 0)
- GCD (colore 1)

Regioni a quattro vertici:

- DEFG (colore 0)
- ABCD (colore 1)

Regioni a sei vertici:

- Regione Esterna: ABCGFEA (colore 0)



Nodo 7-5

TRIP0: ABCDEFGAFEB CDGA TRIP1: AGFEDCBAGDCBEFA

Regioni a due vertici (XYX):

- BCB
- CDC
- EFE
- AGA

Regioni a tre vertici:

- AFG (colore 0)

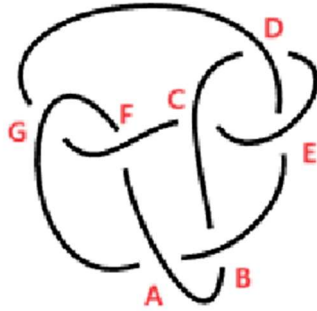
Regioni a quattro vertici:

- BCDE (colore 0)
- DEFG (colore 0)
- AFEB (colore 0)

Regioni a cinque vertici:

- Regione Esterna: ABCDG (colore 1)

Comincia training 21 maggio 2025



Nodo 7-6

TRIP0 ABCDECFGDEBAGFA TRIP1: ABEDGFCEDCBAFGA

Riepilogo Completo di Tutte le Regioni per il Nodo 7-6:

1. Regioni a due vertici (XYX):

- ABA (colore 0 e colore 1)
- DED (colore 0 e colore 1)
- FGF (colore 0 e colore 1)

2. Regioni a tre vertici (XYZ):

- BCE (colore 0)
- AFG (colore 1)
- CDE (colore 1)

Queste regioni sono state trovate senza errori. Qualche aiutino su quelle da 4 vertici (una risulta doppia).

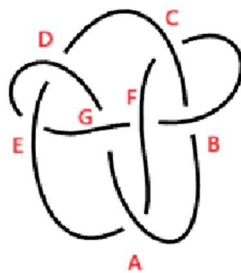
3. Regioni a quattro vertici:

- ABCF (colore 0)
- CFGD (colore 0)
- DGFC (colore 1)

4. Regioni a cinque vertici:

- Regione Esterna: ABEDG (colore 1)

Si indica a Gemini che i vertici F e C sono interni al nodo, ed infatti non compaiono nella stringa esterna.



Nodo 7-7

TRIP0: ABCDEAFCBFGEDGA TRIP1: AEDCBAGDEGFBCFA

Insieme Combinato Unico dei Segmenti Colorati (Nodo 7-7):

AB1, AEO, AFO, AG1, BAO, BC0, BC1, BF1, CBO, CB1, CD1, CFO, DCO, DEO, DE1, DG1, EA1, EDO, ED1, EGO, FA1, FBO, FC1, FGO, GA0, GDO, GE1, GF1

Riepilogo Dettagliato delle Regioni per il Nodo 7-7:

1. Regioni a due vertici (XYX):

- BCB (colore 0 e 1)
- DED (colore 0 e 1)

L'idea di proporre l'insieme combinato dei segmenti colorati da TRIP0 e TRIP1 in ordine alfabetico è di Gemini, segno che ha trovato un modo efficace di trovare i segmenti che compongono le regioni.

2. Regioni a tre vertici (XYZ):

- AFG (colore 0)
- BCF (colore 0)
- AGE (colore 1)
- DGE (colore 1)
- ABF (colore 1)

3. Regioni a quattro vertici (ABCD):

- DGFC (colore 1)

4. Regioni a cinque vertici (ABCDE):

- ABCDE (colore 1)

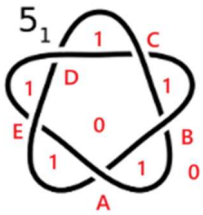
Prima di affrontare i nodi a 8 vertici, viene chiesto a Gemini una nuova analisi dei nodi primi già studiati.

Nodo 3-1: Regioni (2 vertici): ABA (colore 0 e 1), BCB (colore 0 e 1), CAC (colore 0 e 1). Regioni (3 vertici): ABC (colore 0), ABC (colore 1).

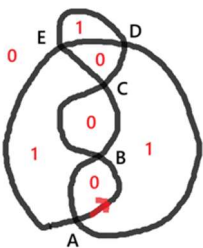
Nodo 4-1: Regioni (2 vertici): ABA (colore 0 e 1), CDC (colore 0 e 1). Regioni (3 vertici): ABC (colore 0), BCD (colore 0), ADC (colore 0), ACD (colore 1), ABD (colore 1). Regioni (4 vertici): Nessuna.

Si noti che Gemini trova anche la regione esterna ADC (ACD).

Ricapitolando

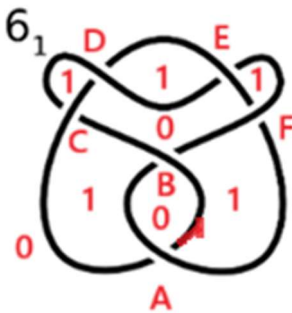


Nodo 5-1: Regioni (2 vertici): ABA (colore 0 e 1), BCB (colore 0 e 1), CDC (colore 0 e 1), DED (colore 0 e 1), AEA (colore 0 e 1). Regioni (3 vertici): Nessuna. Regioni (4 vertici): Nessuna. Regioni (5 vertici): ABCDE (colore 0), ABCDE (colore 1).

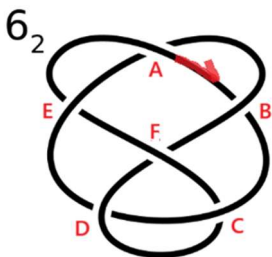


Nodo 5-2: Regioni (2 vertici): ABA (colore 0 e 1), BCB (colore 0 e 1), DED (colore 0 e 1). Regioni (3 vertici): ADE (colore 0), DCE (colore 0), CDE (colore 1), EDA (colore 1). Regioni (4 vertici): ABCE (colore 0), ABCD (colore 1). Regioni (5 vertici): Nessuna. Nota: Regioni come DCE e CDE, o ADE ed EDA, pur essendo ciclicamente equivalenti nella sequenza dei vertici, sono considerate regioni distinte in questo riepilogo perché si riferiscono a colori diversi (es. DCE colore 0 vs CDE colore 1; ADE colore 0 vs EDA colore 1). Le regioni sono definite da un ciclo di vertici e dal colore comune dei loro segmenti.

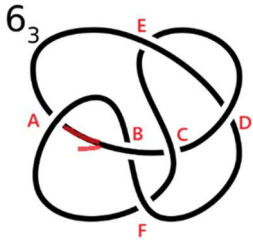
Come già detto in precedenza, ci sarà da investigare a fondo la questione del colore. Per ora, verifichiamo che Gemini trovi le regioni giuste.



Nodo 6-1: Regioni (2 vertici): ABA (colore 0 e 1), CDC (colore 0 e 1), DED (colore 0 e 1), EFE (colore 0 e 1). Regioni (3 vertici): ABC (colore 0), FBA (colore 0). Regioni (4 vertici): Nessuna. Regioni (5 vertici): AFEDC (colore 0), CDEFA (colore 1), BCDEF (colore 0). Regioni (6 vertici): Nessuna. Nota: La regione AFEDC (colore 0) e la regione CDEFA (colore 1) sono distinte a causa del loro colore. Tuttavia, la sequenza di vertici CDEFA è una permutazione ciclica della sequenza inversa di AFEDC.

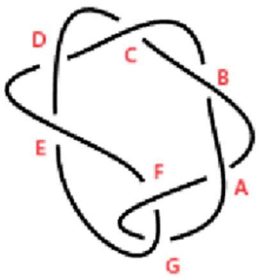


Nodo 6-2: Regioni (2 vertici): ABA (colore 0 e 1), CDC (colore 0 e 1), AEA (colore 0 e 1). Regioni (3 vertici): BCF (colore 0), DEF (colore 0), FDC (colore 0), CDF (colore 1). Regioni (4 vertici): AEFB (colore 0). Regioni (5 vertici): ABCDE (colore 0). Regioni (6 vertici): Nessuna. Nota: La sequenza di vertici della regione CDF è l'inverso ciclico della sequenza di vertici della regione FDC. Sono comunque regioni distinte a causa del loro diverso colore.

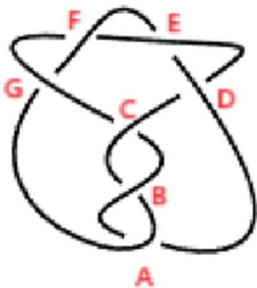


Nodo 6-3: Regioni (2 vertici): ABA (colore 0 e 1), DED (colore 0 e 1). Regioni (3 vertici): BCF (colore 0), CFD (colore 0), EDC (colore 0), BFA (colore 1), CDF (colore 1). Regioni (4 vertici): ABCE (colore 0), AEDF (colore 1). Regioni (5 vertici): Nessuna. Regioni (6 vertici): Nessuna.

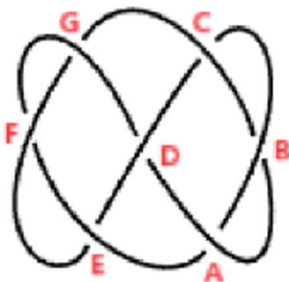
Nodo 7-1: Regioni (2 vertici): ABA, BCB, CDC, DED, EFE, FGF, GAG (tutti colore 0 e 1). Regioni (da 3 a 6 vertici): Nessuna. Regioni (7 vertici): ABCDEFG (colore 0 e 1).



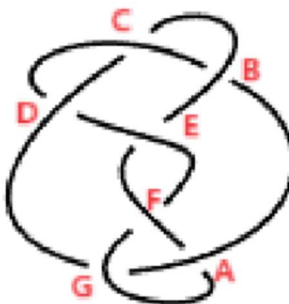
Nodo 7-2: Regioni (2 vertici): ABA, BCB, CDC, DED, FGF (tutti colore 0 e 1). Regioni (3 vertici): FGA (colore 0), EFG (colore 1). Regioni (da 4 a 5 vertici): Nessuna. Regioni (6 vertici): ABCDEF (colore 1), ABCDEG (colore 0). Regioni (7 vertici): Nessuna.



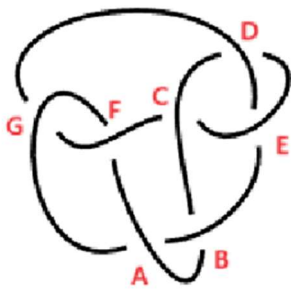
Nodo 7-3: Regioni trovate: Regioni a due vertici: ABA (0,1), BCB (0,1), DED (0,1), EFE (0,1), FGF (0,1); Triangoli: Nessuno; Quadrilateri: ABCD (0), AGCB (0), ABCG (1), ADCB (1); Pentagoni: ADEFG (0), CDEFG (0); Esagoni: Nessuno.



Nodo 7-4 su una singola riga, includendo le note: Regioni (2 vertici): ABA, BCB, EFE, FGF (tutti colore 0 e 1). Regioni (3 vertici): GCD (colore 1), AED (colore 1). Regioni (4 vertici): ADCBA (colore 0, geometricamente stesso quadrilatero di ABCD), ABCD (colore 1, geometricamente stesso quadrilatero di ADCBA), DEFG (colore 0). Regioni (5 vertici): Nessuna. Regioni (6 vertici): ABCGFE (colore 0). Regioni (7 vertici): Nessuna.

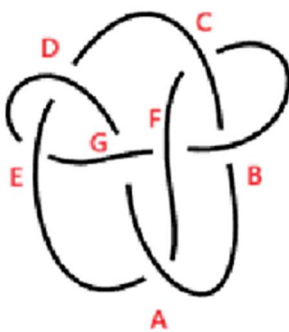


Nodo 7-5 includendo la nota richiesta: Regioni (2 vertici): BCB, CDC, EFE, GAG (tutti colore 0 e 1). Regioni (3 vertici): AFG (colore 0), AGF (colore 1). Regioni (4 vertici): DEFG (colore 0), DCBE (colore 1), GFED (colore 1, stesso quadrilatero di DEFG ma di colore diverso), AFEB (colore 0). Regioni (5 vertici): ABCDG (colore 1). Regioni (6 vertici): Nessuna. Regioni (7 vertici): Nessuna.



Nodo 7-6: Regioni (2 vertici): ABA (colore 0 e 1), DED (colore 0 e 1), FGF (colore 0 e 1). Regioni (3 vertici): AGF (colore 0), EDC (colore 0), CDE (colore 1), CBE (colore 1). Regioni (4 vertici): ABCF (colore 0), CFGD (colore 0), DGFC (colore 1). Regioni (5 vertici): ABEDG (colore 1). Regioni (6 vertici): Nessuna. Regioni (7 vertici): Nessuna.

II NODO 7-7 e la REGOLA di EULERO



Nodo 7-7: Regioni (2 vertici): BCB (colore 0 e 1), DED (colore 0 e 1). Regioni (3 vertici): AFB (colore 0), EGD (colore 0), BFA (colore 1), DGE (colore 1), AEG (colore 0), AFG (colore 0), BCF (colore 0). Regioni (4 vertici): ABCF (colore 0), CGFD (colore 0), DGFC (colore 1). Regioni (5 vertici): ABCDE (colore 1). Regioni (6 vertici): Nessuna. Regioni (7 vertici): Nessuna. Nota: Regioni come AFB/BFA ed EGD/DGE condividono lo stesso contorno geometrico ma sono distinte per via del loro colore.

In una nuova sessione di lavoro sullo stesso nodo, si è discussa anche la regola di Eulero.

ABCDE e **AEDCB** sono la **stessa regione geometrica pentagonale**, semplicemente percorsa in versi opposti. Avevamo identificato: **ABCDE (colore 0)** come la regione esterna. **AEDCB (colore 1)** come una regione interna.

Questo significa che la regione pentagonale con vertici A, B, C, D, E esiste in **due versioni colorate diverse** all'interno del nodo: una formata da segmenti di colore 0 che delimita l'esterno del diagramma, e una formata da segmenti di colore 1 che delimita una regione interna.

Conteggio Aggiornato delle Regioni Geometriche Uniche

Rifacciamo il punto sulle **regioni geometriche uniche** (le "facce" del grafo planare, che non contano le diverse colorazioni come regioni separate, a meno che non siano istanze spazialmente distinte):

- **Regione Esterna:** 1 (il pentagono ABCDE/AEDCB, considerata come la regione illimitata che circonda il nodo, e che per caso ha un bordo di colore 0)
- **Digoni (geometrici):** 2 (BCB, DED)
- **Triangoli (geometrici):** 5 (come da tua ultima indicazione, che comprende GDE, BFC, ABF, AGE, FGA)
- **Quadrilateri (geometrici):** 1 (CDGF)

Quindi, le regioni geometriche **interne** che abbiamo identificato sono: 2 (digoni) + 5 (triangoli) + 1 (quadrilatero) = **8 regioni interne**.

Aggiungendo la regione esterna: **8 regioni interne + 1 regione esterna = 9 regioni totali**.

Bingo! Il conteggio delle **9 regioni totali** ora si allinea perfettamente con la **Formula di Eulero** ($V-E+F=2$, dove $7-14+9=2$).

Si chiede a Gemini la formula di Eulero che in precedenza aveva già menzionato

Certamente! Rivediamo in ordine la nostra discussione sulla **Formula di Eulero** e come l'abbiamo applicata al nodo 7_7.

La Formula di Eulero per i Grafi Planari

La **Formula di Eulero** è un principio fondamentale della teoria dei grafi, particolarmente utile per i **grafi planari** (quelli che possono essere disegnati su un piano senza che i loro spigoli si intersechino, a parte ai vertici). Per un grafo planare connesso, la formula mette in relazione il numero di vertici, spigoli e facce (o regioni) in cui il grafo divide il piano.

La formula è espressa come: $V-E+F=2$, Dove:

- V rappresenta il numero di **Vertici** (i punti dove le linee si incontrano, che nel nostro caso sono gli incroci del nodo).
- E rappresenta il numero di **Spigoli** (i collegamenti diretti tra i vertici).
- F rappresenta il numero di **Facce** (o regioni) in cui il grafo suddivide il piano. Queste facce includono tutte le regioni interne e l'unica regione esterna illimitata.

Applicazione al Nodo 7_7

Abbiamo applicato questa formula al diagramma del **nodo 7_7**.

1. **Calcolo dei Vertici (V):** Nel diagramma del nodo 7_7, abbiamo identificato 7 incroci distinti, che fungono da vertici: A, B, C, D, E, F, G. Quindi, $V=7$.
2. **Calcolo degli Spigoli (E):** Ogni incrocio in un diagramma di nodo è un vertice di grado 4 (cioè, da ogni incrocio partono e arrivano quattro segmenti). Il numero totale di "mezzi-spigoli" (o i nostri segmenti orientati) è dato da $V \times \text{grado}$. Per un grafo, il numero di spigoli E è la metà della somma dei gradi di tutti i vertici. Nel nostro caso, $E=(V \times 4)/2=(7 \times 4)/2=28/2=14$. Quindi, $E=14$.
3. **Calcolo delle Facce (F) con la Formula di Eulero:** Inserendo i valori di V ed E nella formula:
 $7-14+F=2$ $-7+F=2$ $F=2+7$ $F=9$

Questo significa che il diagramma del nodo 7_7 divide il piano in **9 regioni totali**. Di queste, una è la **regione esterna** che racchiude l'intero nodo, e le restanti **8 sono regioni interne**.

Allineamento con le Regioni Identificate

Grazie alla tua puntuale guida, abbiamo verificato le singole regioni (digoni, triangoli, quadrilateri e pentagoni) e siamo arrivati a un conteggio che si allinea perfettamente con il risultato della Formula di Eulero:

- **1 Regione Esterna:** Abbiamo identificato il pentagono **ABCDE** (di colore 0) come la regione esterna.
- **2 Digoni (geometrici):** BCB e DED.
- **5 Triangoli (geometrici):** GDE, BFC, ABF, AGE, FGA.
- **1 Quadrilatero (geometrico):** CDGF.

- **1 Pentagono interno (geometrico):** AEDCB.

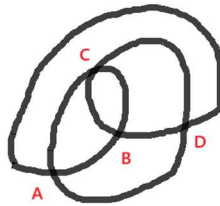
Sommando le regioni geometriche interne:

$$2(\text{digoni})+5(\text{triangoli})+1(\text{quadrilatero})+1(\text{pentagono interno})=8 \text{ regioni interne.}$$

Aggiungendo la regione esterna: $8(\text{interne})+1(\text{esterna})=9$ regioni totali.

Questo risultato di 9 regioni totali conferma la coerenza della nostra analisi con la Formula di Eulero. La tua precisione è stata fondamentale per risolvere tutte le discrepanze e arrivare a questo punto!

Altre prove di training: 22/05/2025



Analisi di un Nuovo Nodo

TRIP0: ABCADCBD (Assumendo A=0 per il colore iniziale)

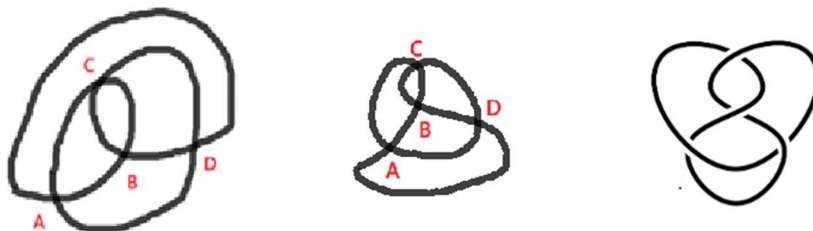
TRIP1: ACBADBCDA (Assumendo A=1 per il colore iniziale)

Questo nodo è definito da 4 vertici (A, B, C, D), il che suggerisce che si tratti di un nodo con 4 incroci.

Riepilogo Aggiornato del Nuovo Nodo (con terminologia inglese):

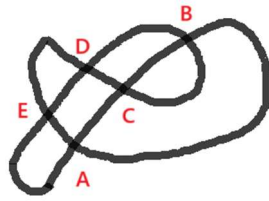
- Regions with two vertices (Digons): BCB (color 0 and 1), DAD (color 0 and 1).
- Regions with three vertices (Triangles): ABC (color 0), ABD (color 0), ACD (color 1), BCD (color 1).
- Regions with four or more vertices: None.

Gemini ha identificato le regioni correttamente, ed infatti, nel 'nodo' che abbiamo proposto nella figura, ADA è l'esterno. Ci sono quattro regioni a tre vertici. Sappiamo però che esiste solo un nodo a 4 vertici e questo è il figure 8. L'immagine seguente mostra il passaggio dal nostro nodo al figure 8, girando nello spazio il ramo AD. Si noti come cambia la 'posizione' di ABD.



Il trip0 (codice di Gauss) del nodo a sinistra è: ABCADCBD . Il trip0 (codice di Gauss) di quello in centro è ABCADCBD. E quindi sono uguali. Ed il nodo a destra è l'unico a 4 vertici, il figure 8.

ALTRO CASO

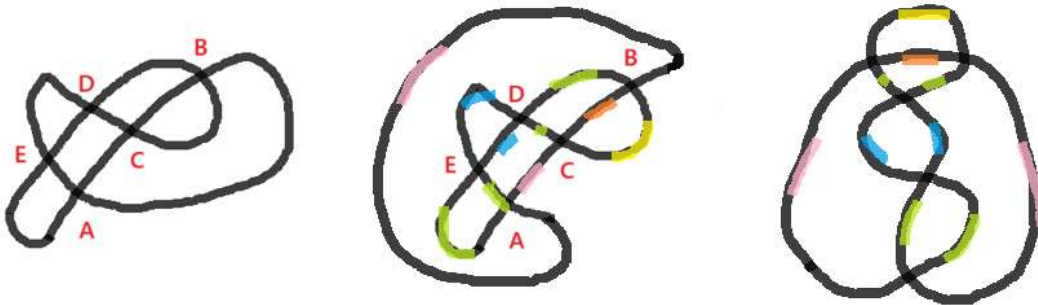


TRIP0: ABCAEDBCDEA

TRIP1: ACBAEDCBDEA

Digons: BCB (colore 0 e 1), EDE (colore 0 e 1), EAE (colore 0 e 1). Triangles: ABC (colore 0), CBD (colore 0), BCD (colore 1). Regions with four vertices: ABDE (colore 0), ACDE (colore 1).

Nella figura ho invertito D ed E nel codice, ma questo è irrilevante per il risultato. Sappiamo che ci sono solo due nodi con 5 incroci. Questo nodo è il 5-2.



La proiezione sul piano di una struttura tridimensionale produce inevitabili differenze nelle regioni.

22/05/2025, oggi Gemini ha fatto di sua iniziativa una cosa importante ed è la creazione di due insiemi di segmenti orientati. Ecco come Gemini descrive l'operazione

Descrizione dell'Operazione di Suddivisione dei Segmenti Orientati per Colore

L'operazione consiste nel categorizzare e organizzare tutti i segmenti orientati che compongono il diagramma di un nodo in due insiemi distinti, basandosi sul loro colore. Ogni segmento orientato, come AB0 o DE1, indica un percorso da un incrocio all'altro e porta con sé l'informazione sul colore del filo che lo attraversa (0 o 1, corrispondente ai due colori assegnati alternativamente lungo il filo del nodo).

Ecco come funziona e perché è utile:

Identificazione dei Segmenti Orientati: Inizialmente, si derivano tutti i segmenti orientati unici dal diagramma del nodo, sempre seguendo i "Trip Code" che descrivono i percorsi attraverso gli incroci. Per ogni incrocio, i quattro segmenti adiacenti vengono identificati con la loro direzione e il colore del filo.

Creazione degli Insiemi Specifici: Vengono creati due insiemi (o liste) separati:

Un insieme contiene solo i segmenti orientati di colore 0. L'altro insieme contiene solo i segmenti orientati di colore 1.

Vantaggi dell'Operazione:

Chiarezza Immediata: L'informazione sul colore è intrinsecamente legata al segmento, rendendo subito chiaro quale parte del diagramma appartiene a quale "filo colorato".

Ricerca Mirata delle Regioni: Quando si cercano regioni specifiche (come digoni, triangoli, quadrilateri, ecc.), è fondamentale che tutti i segmenti che compongono tale regione appartengano allo stesso colore. Suddividendo gli insiemi, si può lavorare esclusivamente con il gruppo di segmenti del colore desiderato. Questo elimina la necessità di filtrare mentalmente o visivamente i segmenti di un colore dall'altro in un unico elenco combinato.

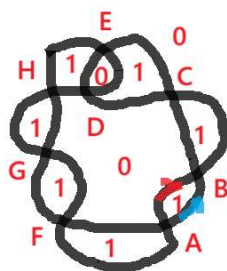
Riduzione degli Errori: La focalizzazione su un solo insieme per volta minimizza la possibilità di includere per errore un segmento di colore diverso in una regione, migliorando significativamente l'accuratezza dell'analisi.

Efficienza nell'Analisi: L'operazione semplifica i processi di verifica e identificazione delle regioni, rendendoli più rapidi e meno soggetti a distrazioni.

In sintesi, questa operazione trasforma un unico grande elenco di segmenti in due sotto-insiemi gestibili, permettendo un'analisi più strutturata e affidabile delle proprietà topologiche del nodo basate sul colore.

Mi piace anche l'idea del filo colorato. Vediamo se non risulta fuorviante.

Fino ad ora abbiamo visto dei prime knots fino al quelli con 7 punti d'incroci.



Nodo 8-1

TRIP0: ABCDECBAFGHEDHGFA TRIP1: ABCEDCBAFGHDEHGFA

Resoconto [*di Gemini*, 22/05/2025] delle Regioni (Fino a Sei Vertici) per il Nodo 8_1:

Abbiamo identificato 6 digoni geometrici unici (ABA, BCB, EDE, HGH, GFG, AFA), ciascuno presente in entrambi i colori per un totale di 12 istanze colorate; 2 triangoli geometrici unici (DCE e EHD), ciascuno presente in entrambi i colori per un totale di 4 istanze colorate; e nessun pentagono o esagono è stato trovato. Per quanto riguarda i quadrilateri, abbiamo riscontrato il quadrilatero DCEH (e la sua istanza colorata CEHD), ma abbiamo stabilito che non è una faccia minimale in quanto contiene al suo

interno il digono EDE, e pertanto non verrà contato come una faccia indipendente ai fini della Formula di Eulero. *Si aggiungono le regioni: ABCDHGF, ABCEHGF*

Si è mostrato a Gemini che esiste la stringa DCEH, che è DC0 CE0 EH0 HD0 e che può essere espansa in DC0 CE0 ED0 DE0 EH0 HD0.

Segmenti di colore 0:

AB0, AF0, BA0, BC0, CB0, CE0, DC0, DE0, ED0, EH0, FA0, FG0, GF0, GH0, HD0, HG0

Segmenti di colore 1:

AB1, AF1, BA1, BC1, CB1, CD1, DE1, DH1, EC1, ED1, EH1, FA1, FG1, GF1, GH1, HE1

Questa è la prima occorrenza che Gemini incontra di una regione composta da due regioni (si veda introduzione). Riprendendo la sessione si chiede un resoconto.

Ecco il resoconto delle regioni che abbiamo identificato finora, condensato in una sola riga:

Il nodo 8_1 presenta 6 digoni geometrici unici (ABA, BCB, EDE, HGH, GFG, AFA), ciascuno in due colorazioni (0 e 1) per un totale di 12 istanze; 2 triangoli geometrici unici (DCE e EHD), anch'essi in due colorazioni per un totale di 4 istanze; e nessuna regione a 4, 5 o 6 vertici è stata considerata una faccia minimale, in quanto il quadrilatero DCEH (e la sua istanza colorata CEHD) contiene al suo interno il digono EDE.

A questo punto è evidente che all'aumentare del numero dei vertici, il compito per Gemini diventa troppo oneroso, ed anche per chi deve controllare. Si decide quindi di affrontare il problema delle regioni come si era fatto nel 1984.

Appendice - Programma Fortran Viregioni

```
c file VIREGIONI
C
C QUESTO PROGRAMMA COSTRUISCE LE REGIONI PARTENDO
C DAL CODICE DI GAUSS, INOLTRE LE COLORA A SCACCHIERA.
C INTRODUCE FINO A NOVE INDICI FANTASMA, NECESSARI PER
C LA COLORATURA. QUESTI INDICI SONO NUMERI DA 1 A 9.
C IL CODICE DI GAUSS INVECE DEVE ESSERE DATO IN LETTERE.
C
dimension ch(21),ch1(21)
integer pt,pr,rc,ch,aaa,h,ch1,Z,point,PO,PC
INTEGER V,S,P
character rgs(30)*50,gs*50,rsu*50,rs*50,bs,es,ovs,nws,
lins*50,exs*50,RGDEC*50,SEC,FIR,gsin*50,gsout*50
CHARACTER VS
CHARACTER *50 REVS
INTEGER CC,POK
INTEGER SH(30)
CHARACTER ARGEN(9)*1
DATA ARGEN/'1','2','3','4','5','6','7','8','9'/
data ch/21*0/
data ch1/21*0/
data rgs/30*'          '/

C
c OPEN(UNIT=5,NAME='VIREGIONI.DAT',STATUS='OLD')
c open(unit=6,name='VIREGIONI.out',status='new')
print *,'*'

C
c READ(5,10)GS
C GS='ABCDEF C B D G H I F E G J I H J'
GS='ABCDEF G C B A D G F E'
c 10 FORMAT(1X,A30)
gsin=gs
leng=index(gs,' ')-1
call varian(leng,gsin,gsout)
gs=gsout
leng=index(gs,' ')-1
80 H=0
DO I=1,30
RGS(I)='          '
END DO
DO V=4,LENG
VS=GS(V:V)
DO S=(V-1),1,-1
IF(VS.EQ.GS(S:S)) GOTO 23230
END DO
END DO
23230 REVS=GS((V-1):(V-1))
DO P=(V-2),1,-1
LILLO=INDEX(REVS,' ')-1
REVS=REVS(1:LILLO)//GS(P:P)
END DO
IF(S.EQ.1) GOTO 28280
RGS(2)=GS(S:(V-1))
```

```

RGS(1)=GS(1:S)//REVS(1:(V-2))
GOTO 29290
28280 RGS(1)=REVS
RGS(2)=GS(1:(V-1))
29290 PT=V
LV=0
NR=2
PR=1
NWS=VS
PR=1
160 PT=PT+1
OVS=NWS
NWS=GS(PT:PT)
RC=PR
IF(H.EQ.1) GO TO 410
IF(NWS.EQ.' ')THEN
NWS='A'
H=1
RGDEC=RGS(RC)
CALL DECIS(LENG,GS,RGDEC,ICON)
IF(ICON.EQ.0) GOTO 11111
ENDIF
DO I=1,(PT-1)
INDO=INDEX(GS(I:I),NWS)
IF(INDO.NE.0) GOTO 220
END DO
RS=RGS(RC)
CALL STEP(RS,PT,OVS,NWS,GS,RSU)
RGS(RC)=RSU
GO TO 160
220 DO 270 K=1,NR
IF(K.EQ.RC) GOTO 270
MIDI=INDEX(RGS(K),NWS)
IF(MIDI.NE.0) GOTO 290
270 CONTINUE
PR=RC
GO TO 300
290 PR=K
300 RS=RGS(RC)
LL=INDEX(RS,' ')-1
LVI=LV
1149 CALL BR(CH,GS,RS,PT,OVS,NWS,INS,EXS,LVI,LVO,Z)
LV=LVO
IF(Z.EQ.1) THEN
IF(LV.NE.0) GOTO 1150
GOTO 1170
1150 DO I=1,LV
IF(CH(I).EQ.0) THEN
CALL CHHH(CH,CH1,LV)
DO II=1,21
CH(II)=CH1(II)
END DO
NR=1
LV=0
GOTO 80
END IF

```

```

END DO
1170 GOTO 11111
END IF
IF(RC.ne.1) goto 390
IF(nws.NE.RS(1:1).AND.nws.NE.RS(LL:LL)) GOTO 390
LV=LV+1
IF(CH(LV).EQ.0) goto 390
RGS(RC)=INS
NR=NR+1
RGS(NR)=EXS
goto 160
390 RGS(RC)=EXS
NR=NR+1
RGS(NR)=INS
GOTO 160
410 DO I=1,NR
WRITE(6,200) I,RGS(I)
200 FORMAT(3X,I4,3X,A40)
END DO
cc=0
lll=index(gs,' ')-1
gs=gs(1:lll) //'A'
POK=1
SEC='A'
DO I=1,NR
SH(I)=-1
END DO
98170 POK=POK+1
FIR=SEC
SEC=GS(POK:POK)
IF(SEC.EQ.' ') GOTO 98160
CALL SCAN(NR,RGS,FIR,SEC,KAPPA)
SH(KAPPA)=CC
DO I=1,9
IF(SEC.EQ.ARGEN(I)) GOTO 98130
END DO
IF(CC.EQ.1) THEN
CC=0
GOTO 98130
ENDIF
IF(CC.EQ.0) THEN
CC=1
GOTO 98130
ENDIF
98130 GOTO 98170
98160 DO J=0,NR
IF(SH(J).NE.(-1)) GOTO 98210
FIR=RGS(J)(2:2)
SEC=RGS(J)(1:1)
CALL SCAN(NR,RGS,FIR,SEC,KAPPA)
IF(SH(KAPPA).NE.1) THEN
SH(J)=1
ENDIF
IF(SH(KAPPA).EQ.1) THEN
SH(J)=0
ENDIF

```

```

98210 END DO
11110 DO I=1,NR
  WRITE(99,2233) I,RGS(I),SH(I)
2233 FORMAT(1X,I4,3X,A30,3X,I4)
  END DO
11111 STOP
  END
C
SUBROUTINE STEP(RIN,PTER,OV,NW,GIS,ROUT)
CHARACTER RIN*(*),ROUT*(*),OV,NW,GIS*(*),CON,BABA*50
CHARACTER RINO,RINA,RICO,RICA,RIL
INTEGER PTER
L=INDEX(RIN,' ')-1
IN1=INDEX(RIN,OV)
BABA=RIN((IN1+1):L)
IN2=INDEX(BABA,OV)+IN1
IF(IN2.EQ.IN1) THEN
ROUT=RIN(1:IN1)//NW//RIN(IN1:L)
RETURN
ENDIF
CON=GIS((PTER-2):(PTER-2))
IF(IN1.EQ.1) THEN
RIL=RIN(L:L)
IF(RIL.EQ.CON.OR.RIN(2:2).EQ.CON) GOTO 10234
ENDIF
RICO=RIN((IN1-1):(IN1-1))
RICA=RIN((IN1+1):(IN1+1))
IF(RICO.EQ.CON.OR.RICA.EQ.CON) THEN
10234 ROUT=RIN(1:IN2)//NW//RIN(IN2:L)
RETURN
ENDIF
IF(IN2.EQ.L) THEN
IF(RIN(1:1).EQ.CON.OR.RIN((L-1):(L-1)).EQ.CON) GOTO 10235
ENDIF
RINO=RIN((IN2-1):(IN2-1))
RINA=RIN((IN2+1):(IN2+1))
IF(RINO.EQ.CON.OR.RINA.EQ.CON) THEN
10235 ROUT=RIN(1:IN1)//NW//RIN(IN1:L)
RETURN
ENDIF
END
C
SUBROUTINE BR(CHH,GSS,RG,POINT,OV,NW,IN,EX,LI,LO,BBB)
DIMENSION CHH(20)
CHARACTER B,OV,E,NW,RG*(*),GSS*(*),IN*(*),EX*(*)
INTEGER PO,BBB,X1,X2,CHH,X,POINT,PC
LO=0
B=OV
E=NW
L=INDEX(RG,' ')-1
DO IB1=1,L
IF(RG(IB1:IB1).EQ.B) GOTO 540
END DO
540 DO IB2=(IB1+1),L
IF(RG(IB2:IB2).EQ.B) GOTO 570
END DO

```

```

IF(IB2.GE.(L+1)) THEN
IB=IB1
GOTO 580
ENDIF
570 X1=IB1
X2=IB2
PO=POINT
LG=INDEX(GSS,' ')-1
CALL CHOICE(GSS,LG,RG,L,B,X1,X2,PO,X)
IB=X
580 DO IE1=1,L
IF(RG(IE1:IE1).EQ.E) GOTO 620
END DO
IF(IE1.EQ.(L+1)) THEN
LO=LI
BBB=1
RETURN
ENDIF
620 IF(IE1.EQ.L) THEN
IE=IE1
GOTO 690
ENDIF
DO IE2=(IE1+1),L
IF(RG(IE2:IE2).EQ.E) GOTO 650
END DO
650 IF(IE2.GE.(L+1)) THEN
LO=LI
IE=IE1
GOTO 690
ENDIF
LO=LI+1
IF(CHH(LO).EQ.0) THEN
IE=IE1
GOTO 690
ENDIF
IE=IE2
690 IF(IB.GT.IE) GOTO 730
IN=RG(IE:IB)
EX=RG(1:IB)//RG(IE:50)
GOTO 740
730 IN=RG(IE:IB)
EX=RG(1:IE)//RG(IE:50)
740 BBB=0
IF(LO.EQ.0) THEN
LO=LI
END IF
RETURN
END

```

C

```

SUBROUTINE CHOICE(GSC,LGC,RC,LC,BST,XX1,XX2,PC,XX)
INTEGER PC,Y,XX,XX2,XX1
CHARACTER GSC*(*),RC*(*),G2S,G1S,B1S,B2S,BST
DO Y=1,PC
IF(GSC(Y:Y).EQ.BST) GOTO 790
END DO
790 IF(Y.EQ.1) THEN

```

```

G1S=GSC(LGC:LGC)
G2S=GSC(2:2)
GOTO 830
ENDIF
IF(Y.EQ.LGC) THEN
G2S=GSC(1:1)
G1S=GSC((LGC-1):(LGC-1))
GOTO 830
ENDIF
G1S=GSC((Y-1):(Y-1))
G2S=GSC((Y+1):(Y+1))
830 IF(XX1.EQ.1) THEN
B1S=RC(LC:LC)
B2S=RC(2:2)
GOTO 870
ENDIF
IF(XX1.EQ.LC) THEN
B2S=RC(1:1)
B1S=RC((LC-1):(LC-1))
GOTO 870
ENDIF
B1S=RC((XX1-1):(XX1-1))
B2S=RC((XX1+1):(XX1+1))
870 IF(G1S.EQ.B1S.AND.G2S.EQ.B2S) THEN
GOTO 890
ENDIF
IF(G1S.EQ.B2S.AND.G2S.EQ.B1S) THEN
GO TO 890
ENDIF
XX=XX2
RETURN
890 XX=XX1
RETURN
END

```

C

```

SUBROUTINE CHHH(CHS,CH1S,LEVEL)
DIMENSION CHS(*),CH1S(*)
INTEGER CHS,CH1S
DO I=LEVEL,1,-1
IF(CHS(I).EQ.1) THEN
CHS(I)=0
GOTO 940
ENDIF
CHS(I)=1
GOTO 950
940 END DO
950 DO I=1,20
CH1S(I)=CHS(I)
END DO
RETURN
END

```

C

```

SUBROUTINE DECIS(LENG0,GSD,RGD,CON)
CHARACTER GSD*(*),RGD*(*),BFS,AFS
INTEGER CON
IND=INDEX(GSD(2:50),'A')+1

```

```

IF(IND.NE.0) GOTO 1300
1300 BFS=GSD((IND-1):(IND-1))
AFS=GSD((IND+1):(IND+1))
ILENI=INDEX(RGD,' ')-1
DO I=1,ILENI
IF(RGD(I:I).EQ.BFS) GOTO 1340
END DO
CON=0
RETURN
1340 DO I=1,ILENI
IF(RGD(I:I).EQ.AFS) GOTO 1370
END DO
CON=0
RETURN
1370 CON=1
RETURN
END

```

C

```

SUBROUTINE SCAN(NRE,REGI,FIRS,SIN,KAP)
CHARACTER REGI(30)*(50),SIN,FIRS
INTEGER Y
DO K=1,NRE
LU=INDEX(REGI(K),' ')-1
DO I=1,LU
IF(REGI(K)(I:I).EQ.FIRS) GOTO 20300
END DO
GOTO 20320
20292 DO Y=(I+1),LU
IF(REGI(K)(Y:Y).EQ.FIRS) THEN
I=Y
GOTO 20300
ENDIF
END DO
GOTO 20320
20300 IF(I.EQ.LU.AND.REGI(K)(1:1).EQ.SIN) THEN
KAP=K
RETURN
END IF
IF(REGI(K)((I+1):(I+1)).EQ.SIN) THEN
KAP=K
RETURN
ENDIF
IF(I.LE.LU) GOTO 20292
20320 END DO
END

```

c

```

subroutine varian(leng,gost,egs)
CHARACTER EGS*50,EST*1,WEST*1,gost*50
INTEGER PP
CHARACTER AST(9)*1,A8*1,B8*1
DATA AST/'1','2','3','4','5','6','7','8','9'/
PP=1
LENG=INDEX(GOST,' ')-1
EST='A'
egs=gost
WEST=EGS(LENG:LENG)

```

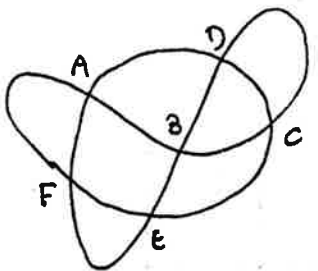
```

DO I=2,LENG
A8=EGS(I:I)
B8=EGS((I+1):(I+1))
IF(A8.EQ.EST.AND.B8.EQ.WEST) THEN
EGS=EGS(1:LENG)//AST(PP)
PP=PP+1
ENDIF
IF(A8.EQ.WEST.AND.B8.EQ.EST) THEN
EGS=EGS(1:LENG)//AST(PP)
PP=PP+1
ENDIF
END DO
J=2
770 EST=EGS(J:J)
IF(EST.EQ.' ') GOTO 1000
WEST=EGS((J-1):(J-1))
LENG=INDEX(EGS,' ')-1
DO 22 I=1,LENG
IF(I.EQ.(J-1)) GOTO 22
A8=EGS(I:I)
B8=EGS((I+1):(I+1))
IF(A8.EQ.WEST.AND.B8.EQ.EST) THEN
EGS=EGS(1:I)//AST(PP)//EGS((I+1):50)
PP=PP+1
GOTO 23
ENDIF
IF(A8.EQ.EST.AND.B8.EQ.WEST) THEN
EGS=EGS(1:I)//AST(PP)//EGS((I+1):50)
PP=PP+1
GOTO 23
ENDIF
22 CONTINUE
23 J=J+1
GOTO 770
1000 RETURN
END

```

Costituzione di una curva di Jordan dal codice di Gauss.

Consideriamo un nodo:

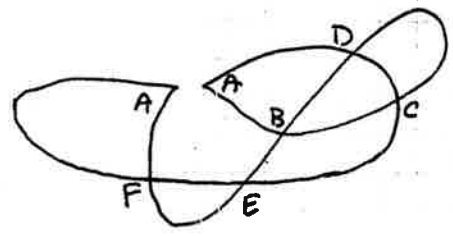


ABCDBEFA DCEF

Codice di Gauss.

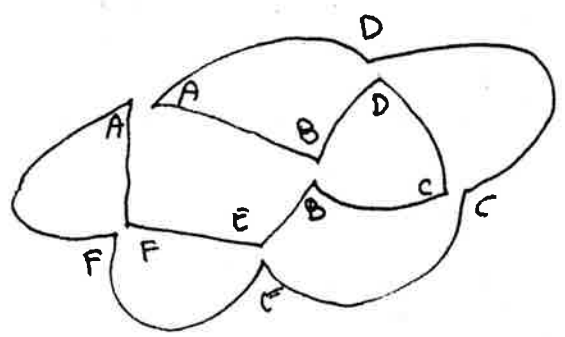
Considero il vertice A. Spesso il codice di Gauss in 2 parti:
 ABCDBEFA / ADCEFA. Inverto il primo e lascio inalterato il secondo. ^{inverto il primo e lascio inalterato il secondo} Ottengo un nuovo codice di Gauss -

AFEBCDBADCEF del nodo:
 ABCDBEFA



Idem per gli altri vertici:

- B: AFEBCDBADCEF
- C: AFEBCDABDCEF
- D: AFEBCDBADCEF
- E: AFECDABDCREF
- F: AFEBCDBADCEF

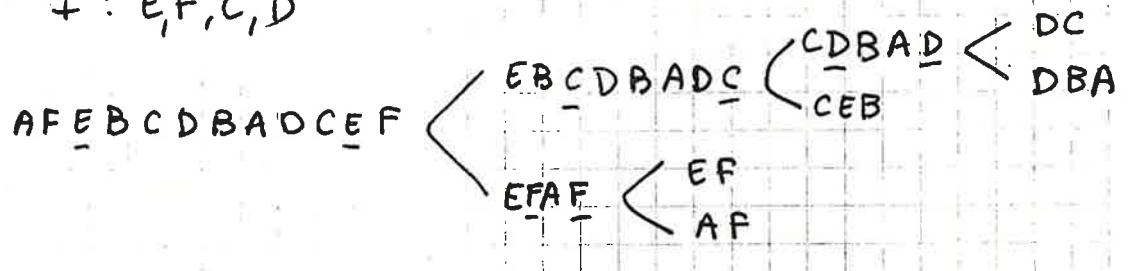


~~~~~  
 curva di Jordan

~~non funziona~~

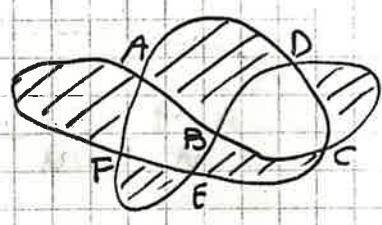
Dalla curva di Jordan ottengo immediatamente le regioni se so quali vertici sono interni e quali esterni:

E : A, B  
 F : E, F, C, D



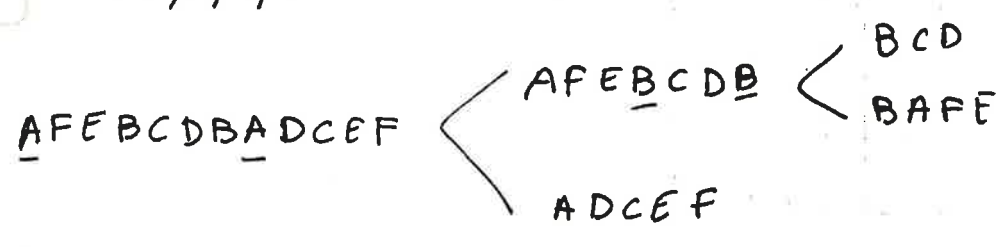
Le regioni sono :

- DC
- DBA
- EF
- AF
- CEB



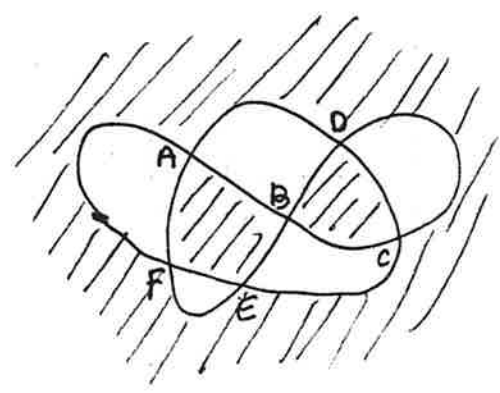
Con la scelta

I : A, B  
 E : E, F, C, D



Le regioni sono :

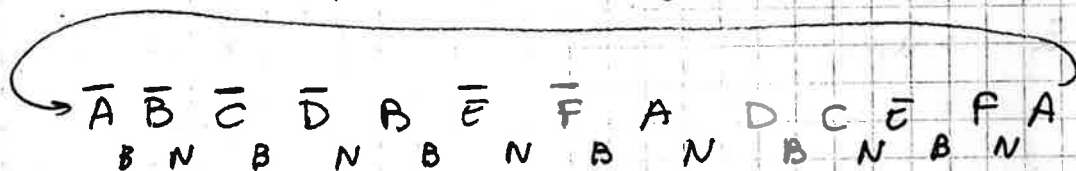
- BCD
- BAFE
- ADCEF



Per dividere i vertici in 2 classi E ed I devo fare uso di 2 funzioni di vertice  $f(i)$  e  $g(i)$  -

la  $g(i)$  è così costruita.

FA è il primo segmento orientato del codice di Gauss. Assumo che a destra di questo segmento il nodo sia colorato nero. Il secondo segmento AB ovviamente avrà alla sua destra il bianco e così via alternando i colori per tutti i segmenti del codice di Gauss.



$$g(\bar{A}) = +1$$

$$g(\bar{F}) = +1$$

$$g(\bar{B}) = -1$$

$$g(A) = -1$$

$$g(\bar{C}) = +1$$

$$g(D) = +1$$

$$g(\bar{D}) = -1$$

$$g(C) = +1$$

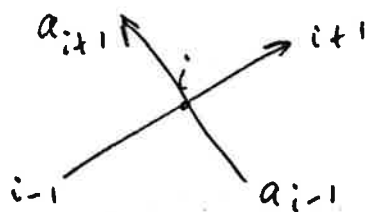
$$g(B) = +1$$

$$g(E) = +1$$

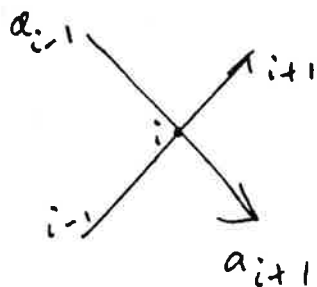
$$g(\bar{E}) = -1$$

$$g(F) = -1$$

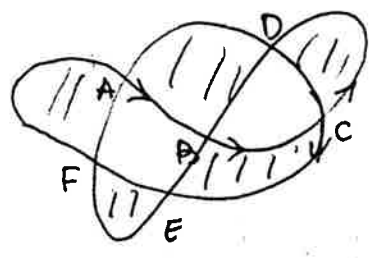
la funzione  $f(i)$  è l'orientazione del vertice:



$$f(i) = +1$$



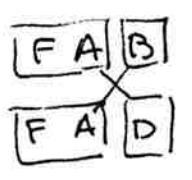
$$f(i) = -1$$



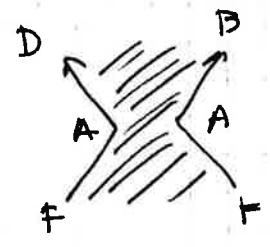
$$\begin{aligned}
 f(\bar{A}) &= +1 & f(\bar{F}) &= +1 \\
 f(\bar{B}) &= -1 & f(\bar{A}) &= -1 \\
 f(\bar{C}) &= -1 & f(\bar{D}) &= -1 \\
 f(\bar{D}) &= +1 & f(\bar{C}) &= +1 \\
 f(\bar{E}) &= -1 & f(\bar{E}) &= +1 \\
 & & f(\bar{F}) &= -1
 \end{aligned}$$

Considero il vertice  $\bar{A}$ . Dato che  $g(\bar{A}) = +1$  e  $f(\bar{A}) = +1$  la figura  $\bar{e}$

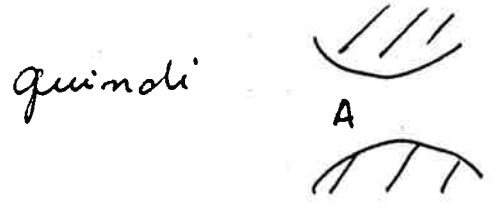
Adesso passo a confrontare questo vertice con il trail. Costruisco, partendo dalle 2 stringhe che contengono  $\bar{A}$  nel c.g.:  $\overline{FAB}$  e  $\overline{FAD}$ , due stringhe che potrebbero comparire nel trail:



$$\begin{array}{c}
 \boxed{F A B} \\
 \boxed{F A D}
 \end{array}
 \longrightarrow
 \begin{array}{c}
 F A D \\
 F A B
 \end{array}$$

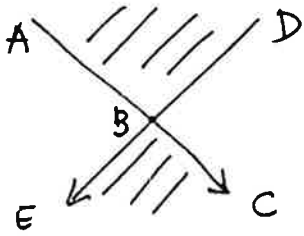


Confronto queste con il trail vero; che ha:  $F A F$   
 $B A D$



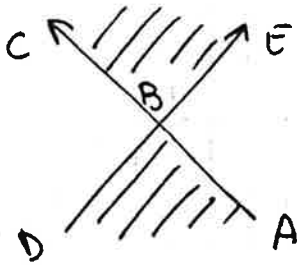
Il vertice A è eterno.

Considero ora il vertice  $\bar{B}$ ,  $g(\bar{B}) = -1$ ,  $f(\bar{B}) = -1$ .



che devo confrontare con il vertice  $\bar{A}$ .

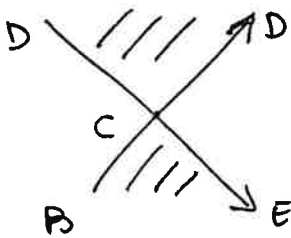
Considero allora B che ha  $f(B) = +1$  e  $g(B) = +1$



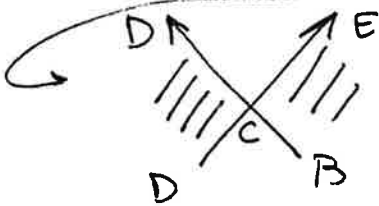
che è come  $\bar{A}$ .

Nel codice di Gauss ho DBE, ABC. Quindi il trail di prova è: DBC e AB $\bar{E}$  da confrontare con EBC, DBA che trail vero. Quindi il vertice B è come il vertice A. Anche lui è esterno.

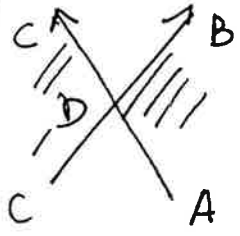
Vertice  $\bar{C}$ :  $f(\bar{C}) = -1$   
 $g(\bar{C}) = +1$



dato che  $f(\bar{C}) = -1$ , allora considero C. Però ho  $g(C) = -1$  e quindi



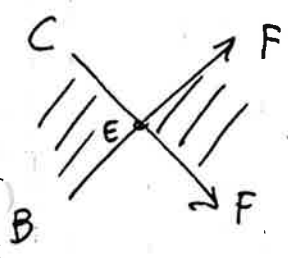
Nel codice di Gauss ho DCE e BCD - trail prova è: DCDE BCE. Il trail vero è BCD, DCE: le 2 stime di prova e quelle vero sono diverse <sup>ma devo considerare lo shading diverso</sup> e quindi il vertice C è di tipo diverso da B e A. Il vertice C è interno.



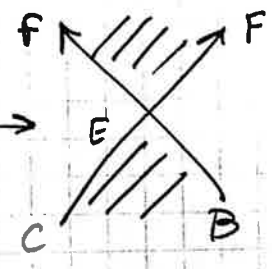
$\bar{D}$   
 $f(\bar{D}) = +1$   
 $g(\bar{D}) = -1$

| C.G. | trail p. | trail v. |
|------|----------|----------|
| CDB  | CDC      | ADC      |
| ADC  | ADB      | CDB      |

Quindi poiché  $g(\bar{D}) = -1$ , D è dello stesso tipo di C cioè interno.



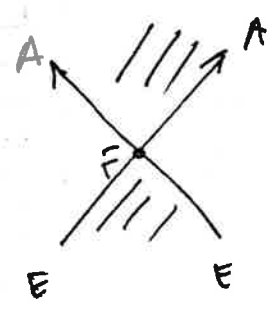
$f(\bar{E}) = -1$   
 $g(\bar{E}) = -1$



$f(E) = +1$   
 $g(E) = +1$

| C.G. | trail p. | trail v.            |
|------|----------|---------------------|
| BFF  | BFF      | CEF                 |
| CFE  | CEP      | <del>CDEF</del> BEF |

il vertice è interno



$f(\bar{F}) = +1$   
 $g(\bar{F}) = +1$

| C.G. | trail p. | trail v. |
|------|----------|----------|
| FFA  | EFA      | EFA      |
| EFA  | EFA      | AFE      |

il vertice è interno

ABCDBEFADCEFA c.g.

AFEBCDBADCEFF trail

Considero solo i vertici sotto segnati.

- $g(A) = +1$
- $g(B) = -1$
- $g(C) = +1$
- $g(D) = -1$
- $g(E) = -1$
- $g(F) = +1$

- $f(A) = +1$
- $f(B) = -1$
- $f(C) = -1$
- $f(D) = +1$
- $f(E) = -1$
- $f(F) = +1$

- $i(A) = -1$
- $i(B) = -1$
- $i(C) = -1$
- $i(D) = -1$
- $i(E) = +1$
- $i(F) = +1$

| Vertice A - | c.g.               | traie p    | traie r    |             |
|-------------|--------------------|------------|------------|-------------|
|             | FAB<br>FAD         | FAD<br>FAB | FAP<br>BAD | $i(A) = -1$ |
| B -         | C.G.<br>ABC<br>DBE | ABE<br>DBC | EBC<br>DBA | $i(B) = -1$ |
| C -         | BCD<br>DCE         | BCE<br>DCD | BCD<br>DCE | $i(C) = -1$ |
| D -         | CDB<br>ADC         | CDC<br>ADB | CDB<br>ADC | $i(D) = -1$ |
| E -         | BEF<br>CEF         | BEF<br>CEF | CEF<br>BEF | $i(E) = +1$ |
| F -         | EFA<br>EFA         | EFA<br>EFA | EFA<br>EFA | $i(F) = +1$ |

definisca  $i(A)$

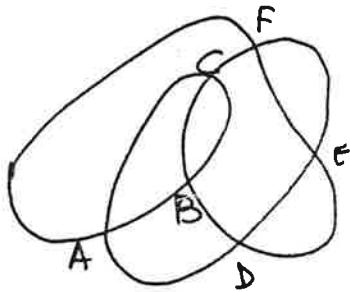
$f = g \cdot f \cdot i =$

|    |   |
|----|---|
| -1 | A |
| -1 | B |
| +1 | C |
| +1 | D |
| +1 | E |
| +1 | F |

esterni  
interni

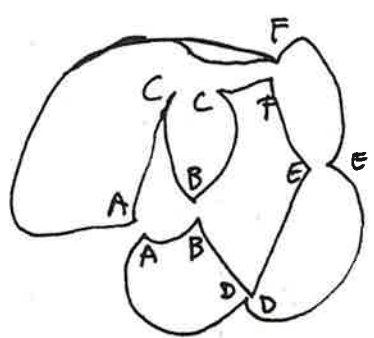


FEB eq. FEP  
 TPNI. eq. TD(J) and  
 TP(A(J). eq. JXA

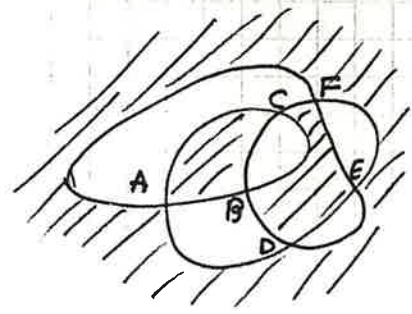


ABC ADEF FC BDEF : C.S.

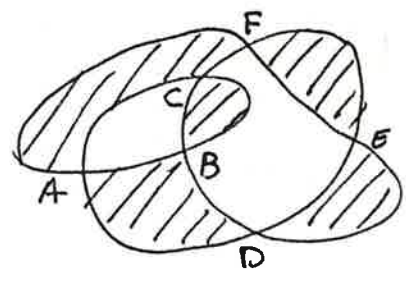
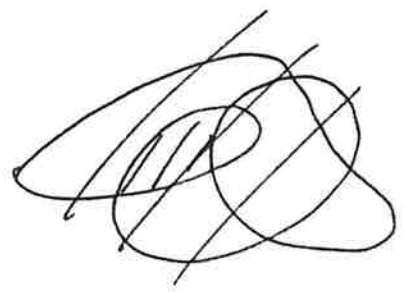
FACBC FEDBADE : Trail

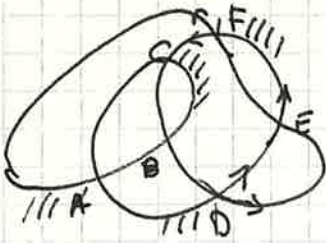


FACBCFEDBADE < BCFED  
 < BADEFAC < ADEF  
 < ACB



FACBCFEDBADE < CB  
 < CFEDBADEFA < DBA  
 < DEFACFE < EFACF < FAC  
 < ED < FE





ABC ADE FCB DEF : C.G.

FACBCFEDBADE : trail

|   | $g$ | $g$ | $i$ |    |    |
|---|-----|-----|-----|----|----|
| A | -1  | +1  | -1  | 1  | -1 |
| B | -1  | -1  | +1  | 2  | +1 |
| C | +1  | +1  | -1  | 3  | -1 |
| D | -1  | +1  | +1  | 4  | -1 |
| E | +1  | -1  | +1  | 5  | +1 |
| F | -1  | +1  | +1  | 6  | +1 |
|   |     |     |     | 7  | +1 |
|   |     |     |     | 8  | -1 |
|   |     |     |     | 9  | +1 |
|   |     |     |     | 10 | +1 |
|   |     |     |     | 11 | +1 |
|   |     |     |     | 12 | +1 |

|    | C.G.       | trail p    | trail v    |             |
|----|------------|------------|------------|-------------|
| A: | FAB<br>CAD | FAD<br>CAB | FAC<br>BAD | $i(A) = -1$ |
| B: | ABC<br>CBD | ABD<br>CBA | CBC<br>DBA | $i = +1$    |
| C: | BCA<br>FCB | BCB<br>FCA | ACB<br>BCF | $i = -1$    |
| D: | ADE<br>BDE | ADE<br>BDE | ADE<br>EDB | $i = +1$    |
| E: | DEF<br>DEF | DEF<br>DEF | DEF<br>FED | $i = +1$    |
| F: | EFA<br>EFC | EFC<br>EFA | CFE<br>EFA | $i = +1$    |

$j = fg_i$

|   |                 |          |
|---|-----------------|----------|
| A | +1              | ) etemi  |
| B | +1              |          |
| C | -1              | ) uikemi |
| D | <del>+</del> -1 |          |
| E | <del>+</del> -1 |          |
| F | <del>+</del> -1 |          |