Università degli Studi di Milano
Dipartimento Scienze dell'Informazione

Thesis progress report

22 Novembre 2000

"On the power of classes of splicing systems"

DOTTORANDA (XIII ciclo): Rosalba Zizza
RELATORE: Prof. Giancarlo Mauri

1 Premessa

Il mio lavoro di tesi è incentrato sullo studio di alcuni modelli non convenzionali di calcolo, ispirati dalla biologia (splicing systems). In queste pagine descriverò lo stato di avanzamento della mia tesi, pertanto, essendo un aggiornamento, non ho ritenuto necessario riscrivere quanto già inserito nella proposta (bibliografia, approccio e descrizione formale dei modelli presentati). Inoltre non farò riferimento a lavori collaterali legati alla teoria dei codici a lunghezza variabile, mia precedente area di ricerca; argomentazioni in tal senso (risultati ottenuti e in corso), alcuni dei quali già presentati nella proposta, saranno direttamente inseriti in un capitolo della tesi. Nelle prossime sezioni introdurrò brevemente il Calcolo Molecolare, area in cui si colloca la mia ricerca attuale, le problematiche generali più rilevanti e quelle da me considerate, e solo relativamente a queste ultime presenterò un breve sunto dello stato dell'arte. La parte centrale di questo documento conterrà una descrizione dei risultati finora ottenuti. Una sezione dedicata ai prossimi punti da sviluppare, ai contributi e un indice provvisorio della tesi concluderà la presentazione.

2 Collocazione

Sebbene la possibilità di usare DNA e tools biologici per formulare e risolvere problemi informatici possa sorprendere, in realtà l'idea soggiacente il Calcolo Molecolare è l'osservazione dell'analogia tra processi biologici e processi informatici. I primi sono il risultato dell'applicazione di semplici operazioni (riconoscimento, copia, splicing, ...) ad informazione codificata in sequenze di DNA, i secondi sono il risultato dell'applicazione di una funzione calcolabile (combinazione di funzioni semplici) ad un certo argomento.

Il DNA (acido deossiribonucleico) è il mezzo per trasferire l'informazione genetica. Leonard Adleman realizzò che era possibile usare DNA per memorizzare l'informazione e gli enzimi per simulare la computazione; nel 1994 ha provato questi discorsi, portando in laboratorio un problema NP-completo (Cammino Hamiltoniano) e risolvendone una istanza. Questa è comunemente considerata la nascita della WET Molecular Computing, per evidenziare come gli informatici entrano in laboratorio per realizzare algoritmi in provetta. Questo lavoro ha aperto la strada ad altri esperimenti su algoritmi per risolvere istanze di problemi NP-completi, nell'ambito della teoria dei grafi (Road Coloring Problem, per esempio), o anche usando tecniche di programmazione dinamica. Infatti, il Calcolo Molecolare ha la potenzialità per superare la performance degli attuali calcolatori, usando ricerche massivamente parallele (esaustive), fornendo vantaggi in termini di velocità del calcolo, di efficienza energetica, di capacità di memorizzazione dell'informazione. In WET Molecular Computing le bio-operazioni sono usate per scrivere programmi, il cui input è un insieme di test tubes contenente strands di DNA e l'output è un test tube con la soluzione alla questione. L'ovvio vantaggio è che si ha già una implementazione in laboratorio. Ciò che impedisce l'automazione a larga scala di tali processi è legato sia al peso della soluzione che teoricamente potrebbe essere tale da rendere praticamente impossibile l'esperimento, sia alla facilità di produzione di errori nei processi di ricombinazione del DNA. Da qui la necessità di definire opportune codifiche del DNA tali da garantire la minima probabilità di errore e interessanti filoni di ricerca sono stati sviluppati in tal senso.

Sebbene proposte di lavoro siano state già fatte per la WET Molecular Computing (cfr. proposta di tesi), mi sono concentrata invece su un altro aspetto del Calcolo Molecolare, siglato come DRY, in cui tutte le bio-operazioni di ricombinazione molecolare sono realizzate da enzimi e sono tradotte in un modello teorico (Splicing System). Introdotti da Tom Head (1987), gli splicing systems garantiscono l'universilità del calcolo, sebbene l'attuale biotecnologia non sia tale da consentire una pratica implementazione. Esistono numerose varianti e definizioni che sono state introdotte in letteratura, ma ci occuperemo solo di alcune di esse.

3 Stato del lavoro: stato dell'arte, proposte fatte e risultati

Come detto, la teoria degli Splicing Systems è stata largamente esplorata e così molti modelli sono stati presentati. Qui accennerò solo quanto necessario per rendere tali pagine self-contained. Prima di ricordare la definizione di splicing, descriviamo brevemente il processo biologico di cut and paste, chiamato *splicing*.

Gli enzimi di restrizione sono proteine capaci di riconoscere uno specifico pattern all'interno di una molecola di DNA e di tagliare tale molecola in un punto specifico all'interno del pattern, spezzando così la molecola originaria in due frammenti. Siccome la molecola di DNA è double-stranded, tale taglio può essere netto o tale da lasciare sporgenze di diversa lunghezza tra i due filamenti di DNA (blunt or sticky ends). Gli enzimi di ligasi attaccano due frammenti di DNA, se questi presentano una opportuna struttura chimica. Pertanto la presenza di tali enzimi in una soluzione acquea con DNA, provoca la generazione di nuove molecole ottenute dalla fase di taglio (prodotta dagli enzimi di restrizione) e dalla fase di incollaggio (ligase).

E' alquando naturale pensare di tradurre il meccanismo ricombinante, appena descritto, nell'ambito dei linguaggi formali, ossia come meccanismo generativo tra parole. Innanzitutto due casi sono da distinguere, tenendo conto che il DNA è presente sia in forma circolare sia in forma lineare (entrambi i casi sono stati affrontati nella tesi). Inoltre, esistono delle varianti di modelli introdotte per simulare il comportamento di una Macchina di Turing (sistemi splicing estesi, con molteplicità...), fornendo così modelli universali. Sia nel caso lineare che circolare abbiamo invece analizzato un modello semplice che opera solo su oggetti finiti. Infatti, sebbene tale condizione sembra restrittiva, nel senso che non è possibile raggiungere l'universalità, essa risulta più realizzabile, dato che molti modelli universali sono praticamente irrealizzabili in laboratorio. Ad esempio, si può intuire che un meccanismo di ricombinazione molecolare è più probabile se ho molte copie delle molecole, piuttosto che un piccolo e preciso numero di esse (significa introdurre la molteplicità nel modello di splicing), perché così le molecole hanno maggiore probabilità di "incontrarsi". Supporremo perciò che prima di effettuare lo splicing tra due molecole, queste sono copiate, così da avere sempre una copia delle molecole per successivi riutilizzi. Inoltre, sebbene sia stato modellato il DNA bidimensionale (double-stranded), ed esistano collegamenti con i dominoes languages, noi abbiamo seguito altre definizioni in cui il DNA è unidimensionale (sigle stranded) e questa assunzione non sembra restrittiva.

In generale, un sistema splicing è una tripla $S = (A, I, R)$, dove A è un alfabeto finito, I è un linguaggio (formale o circolare), detto *iniziale* (rappresenta l'insieme iniziale delle molecole in provetta) e R è un insieme di regole o stringhe speciali (simulano l'azione degli enzimi). Si può definire $L(S)$ il linguaggio splicing generato da S come l'insieme delle parole ottenute per iterazione di R su I e poi sulle parole via via ottenute. Consideriamo sia il caso in cui stiamo modellando lo splicing tra molecole lineari di DNA, sia il caso di molecole circolari.

3.1 Caso lineare

Presentiamo solo la definizione di splicing data da Paun (sono noti tre modelli di splicing che si differenziano per la formalizzazione data delle regole e dunque dell'operazione di splicing). Le regole $r = u_1|u_2\$u_3|u_4 \in R$, ($u_1u_2$ and u_3u_4 sono parole in A^* chiamate *splicing sites* di r) sono parole sull'alfabeto $A \cup \{|, \$\}$, dove $A \cap \{|, \$\} = \emptyset$, e tali che se $w = w_1u_1u_2w_2$ e $v = v_1u_3u_4v_2$, l'applicazione di r a w e v produce due nuove sequenze $x = w_1u_1u_4v_2$ e $z = v_1u_3u_2w_2$. Naturalmente I, R possono essere linguaggi qualsiasi e se denotiamo $H(F_1, F_2)$ la classe dei linguaggi splicing generati da sistemi splicing $S = (A, I, R)$ con $I \subseteq A^*$ linguaggio nella classe F_1 e R nella classe F_2 (F_1, F_2 possono essere finito, regolare, context-free, ...), una domanda immediata può essere determinare la classe dei linguaggi che possono essere generati per splicing, al variare di F_1 e F_2. Alcuni tra i risultati più interessanti sono quelli che provano $H(Fin, Fin) \subset Reg$, $H(Reg, Fin) = Reg$ e $Fin \subset H(Fin, Reg) \subset RE$.

Proposta e risultati. Abbiamo affrontato un problema aperto e alquanto difficile che sta nel decidere, dato un linguaggio regolare L, se L è generabile da splicing finito, ossia se esiste un sistema splicing S con I e R finiti, tali che L è esattamente il linguaggio $L(S)$ generato dal sistema (stiamo, in altre parole, decidendo se L appartiene o no a $H(Fin, Fin)$). Abbiamo risolto tale questione, usando una tecnica che permette di esprimere i cammini di accettazione di un automa solo in termini di alcuni cammini (detti *canonici*), così da esprimere il linguaggio accettato dall'automa solo in termini di linguaggi opportunamente associati a questi cammini. Inoltre, da questi cammini canonici abbiamo descritto come estrarre il sistema, ossia I e R, finiti, tali che $L(S)$ è esattamente il linguaggio accettato dall'automa. Un test solo su questi cammini, infine, permetterà di distinguere gli automi che accettano linguaggi splicing finiti da quelli che non accettano linguaggi splicing finiti. L'idea che abbiamo avuto è stata quella di usare le regole del sistema per ampliare i cicli dell'automa, in quanto solo un numero finito di parole può essere inserito nel linguaggio iniziale. Perciò le regole dovranno interessare i cicli, dovranno essere tali da consentirne l'ampliamento, ma inoltre dovranno garantire la consistenza del sistema: se una regola è applicata a due parole del linguaggio, le parole prodotte per splicing devono ancora appartenere al linguaggio [1, 2]. Abbiamo affrontato, sempre nelle ipotesi di finitezza del sistema, l'analisi del confronto tra le tre definizioni di splicing. Fornendo un linguaggio generabile con sistemi di Paun ma non di Head e un linguaggio generabile con sistema di Pixton ma non di Paun, abbiamo mostrato che la capacità generativa non è la stessa [6, 7].

3.2 Caso circolare

Lo studio delle parole circolari è un classico argomento della teoria dei linguaggi. Infatti, si può definire la relazione di coniugazione $\sim$ in A^*, per la quale $xy \sim yx$, $x, y \in A^*$. Diremo che due parole w, w' sono coniugate se $w \sim w'$, ossia se $w = xy$ e $w' = yx$. Tale relazione è una relazione di equivalenza e denotiamo $A^\sim = A^*/\sim$ l'insieme delle classi di equivalenza rispetto alla coniugazione. Una classe (elemento di $A^\sim$) è chiamata parola circolare, ed è denotata $^\sim w$, $w \in A^*$. Dunque, se consideriamo il morfismo canonico da A^* in $A^\sim$, vuol dire che ad una parola $w \in A^*$ possiamo associare la parola circolare $^\sim w$ (ossia la circolarizzazione di w) e a $^\sim w$ posso associare $w' \in A^*$ tale che $^\sim w' =^\sim w$. Chiaramente il morfismo si può estendere ai linguaggi, associando a $L \subseteq A^*$, $Cir(L) = \{^\sim w | w \in L\}$ la circolarizzazione di L, ossia l'insieme di tutte le parole circolari che corrispondono agli elementi di L, mentre a $C \subseteq A^\sim$ associo $L \subseteq A^*$ tale che $Cir(L) = C$, ossia una linearizzazione di C. Se considero tutte le parole di A^* che corrispondono agli elementi di C, ottengo la *full linearization* di C, scritta $Lin(C)$. Quindi, piuttosto che considerare linguaggi circolari considereremo linguaggi formali chiusi per coniugazione, e siccome un linguaggio circolare è regolare se e solo se la sua full linearization è regolare, considereremo linguaggi formali chiusi per coniugazione e regolari.

E' possibile definire in modi diversi lo splicing circolare, sebbene considereremo sempre un insieme finito di molecole circolari iniziali e un insieme finito di regole. Il fenomeno biologico è il seguente: gli enzimi di restrizione riconoscono un pattern, "aprono" la molecola in un punto specifico all'interno del pattern riconosciuto (linearizzazione). Si possono verificare due comportamenti diversi degli enzimi di ligase: possono semplicemente concatenare i frammenti e circolarizzare la molecola ottenuta (modello formalizzato da Head e Paun) oppure prima di concatenare i frammenti, gli enzimi sostituiscono il pattern riconosciuto (fenomeno modellato da Pixton). Senza soffermarmi sulle definizioni formali, possiamo anche qui definire un sistema splicing circolare e considerare $C(F_1, F_2)$ la classe dei linguaggi circolari generati da un sistema splicing circolare con $I \in F_1$ e $R \in F_2$. E' stato provato che, esistono dei circolari regolari non contenuti in $C(Fin, Fin)$, ma che esistino dei context-free circolari contenuti in $C(Fin, Fin)$ (usando la definizione di Head o Paun). Esistono dei risultati di chiusura solo però considerando ipotesi aggiuntive su R.

Proposta e risultati. Ci siamo inizialmente concentrati sullo studio della definizione di Head e Paun, perchè si avvicinava maggiormente ad un'idea di concatenazione di parole circolari [3], senza considerare ipotesi aggiuntive. Usando questo modello semplificato, abbiamo definito classi di linguaggi regolari e chiusi per coniugazione generabili e non generabili. In [4] abbiamo prima studiato una classe di linguaggi regolari e chiusi per coniugazione legata ai codici (group codes) e abbiamo mostrato che sono generabili. Poi abbiamo esteso la classe considerando linguaggi regolari e chiusi per coniugazione della forma X^*, con X regolare e con alcune ipotesi abbiamo provato la generabilità (fingerprint closed star languages). Abbiamo costruito anche linguaggi di forma diversa generabili (cyclic languages) [5]. Si è anche provata una gerarchia generativa tra le tre definizioni.

4 Piano di lavoro per l'ultimo anno

Rispetto al caso lineare, resta da completare il confronto tra le definizioni di sistemi di splicing, sempre nel caso finito, facendo riferimento ad un'altra definizione data da Head, sebbene poco usata, e confermare un bound stretto sulla complessità dell'algoritmo di decisione proposto.

Per il caso circolare, poiché il potere generativo dei sistemi splicing di Pixton è maggiore degli altri, stiamo lavorando su questa definizione, motivati dal fatto che finora non sono stati prodotti esempi di linguaggi circolari regolari che non siamo generabili da Pixton [5]. Le prove di questi risultati sono abbastanza tecniche e basate soprattutto su combinatoria delle parole. Resta ancora da comprendere il ruolo delle ipotesi aggiuntive introdotte, in alcuni modelli, sull'insieme delle regole, in quanto tali ipotesi garantiscono risultati di chiusura. Lo scopo finale (a lungo termine) che ci si è prefissi è la caratterizzazione dei circolari regolari, attualmente non nota, attraverso lo splicing finito definito da Pixton, sebbene questo risultato sembri alquanto arduo da raggiungere. Per far questo stiamo procedendo infatti a piccoli passi, essendo non semplice la trattazione dei linguaggi circolari. Si era proposto anche lo studio dello splicing circolare usando le Lyndon words. Inoltre l'uso dei plasmidi, DNA circolare, per applicazioni pratiche di algoritmi molecolari, rende anche da un punto di vista WET, interessante e motivante lo studio dello splicing circolare.

Infine ho appena iniziato a studiare altri modelli in Calcolo Molecolare, legati al fenomeno del self-assembly (ibridazione), come gli sticker languages, e anche problematiche in complessità computazionale, approcciate e risolte con questo processo di ricombinazione. Penso, infatti, di orientare la ricerca futura anche in tale direzione.

5 Indice provvisorio della tesi

Riporto nel seguito come intendo organizzare la tesi. Si tratta di un indice generale, non sviluppato completamente in tutte le sottosezioni, soprattutto relativamente alla presentazione dei risultati (cap. 3.3, 3.4, 3.6, 3.7, 3.8, A.3).

1 Introduction

2 WET Molecular Computing

2.1 Adleman's approach

Riferimenti bibliografici

[1] P. Bonizzoni, R. Zizza, *A characterization of regular splicing languages*, Tech. Rep. 254-00 DSI, (2000), 17 pages.

[2] P. Bonizzoni, R. Zizza, *Deciding whether a regular language is a splicing language*, in preparation.

[3] P. Bonizzoni, C. De Felice, G. Mauri, R. Zizza, *On the linear and circular splicing*, WORDS99, (1999), 12 pages.

[4] P. Bonizzoni, C. De Felice, G. Mauri, R. Zizza, *DNA and circular splicing*, Proc. of DNA6 (LNCS volume), 2000, to appear

[5] P. Bonizzoni, C. De Felice, G. Mauri, R. Zizza, *Circular splicing and regularity*, in preparation.

[6] P. Bonizzoni, C. Ferretti, G. Mauri, R. Zizza, *Separating some splicing models*, Proc. of GrammarSystems 2000.

[7] P. Bonizzoni, C. Ferretti, G. Mauri, R. Zizza, *Separating some splicing models*, 2000, submitted.