Universitat Politècnica de Catalunya
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Un Mètode Algebraic per a la Construcció de Cicles Sobre Grafs de deBruijn i Aplicacions a la Codificació Angular

Borja de Balle Pigem

Projecte Final de Carrera
Enginyeria en Telecomunicació (Pla 92)

Co-directors :
Josep M. Fuertes Armengol (ESAII)
Enric Ventura Capell (MA3)

Barcelona, desembre de 2007

Universitat Politècnica de Catalunya
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

Un Mètode Algebraic per a la Construcció de Cicles Sobre Grafs de deBruijn i Aplicacions a la Codificació Angular

Borja de Balle Pigem

Projecte Final de Carrera
Enginyeria en Telecomunicació (Pla 92)

Co-directors :
Josep M. Fuertes Armengol (ESAII)
Enric Ventura Capell (MA3)

Barcelona, desembre de 2007

Agraïments

Vull agrair l'ajuda de totes les persones que han contribuït, d'una manera o altra, a què aquest projecte arribi a bon port. I concretament, a ...

A tota la meva família, especialment a la meva mare i a la meva germana, per tot l'amor i el suport que m'han transmès. I a la Carol, és clar.

Als meus co-directors, sense els quals aquest treball no hagués estat possible, en Pep i l'Enric.

Als eterns dels dinars, converses i *fiestons*: Ricard, Roger, Txema, Gemma, Carla i Laura. I a l'Esteve, per la companyia i per la foto (Figura 1.3).

Als companys de matemàtiques, per fer que aquesta carrera sigui tan divertida. Ells són en Domènec, l'Agnès, en Pau, les Gemmes, les Laures, la Judit, la Glòria, la M. Paz i també en Marcel.

Als telecomats, molt bona gent. Juanjo, Javi, Borja, Pino, Adrià, Ignasi, Oriol, Màxim i Santi.

A les companyes de telecos i els companys de pis, per les bones estones passades en bona companyia. Sandra, Anna, Marc, Lluís, Neus, Nacho, Dídac i Onjo.

I *last but not least*, als wekes d'Olot: Page, Laura, Kuru, Jaume, Ari, Jordi, Sita, Jesús, Marc, Carme, Miquel, David, Lluís, Xevi, Carla i Albert.

Índex

Índex de figures

Índex de taules

Resum

Un codificador angular és un dispositiu electrònic que, acoblat a un eix giratori, ens en permet conèixer la posició angular amb una certa precisió. La llista de màquines que incorporen entre les seves peces un codificador angular és àmplia i diversa. En aquesta llista podem trobar-hi desde satèl·lits i telescopis de llarg abast fins a joguines o reproductors de CD. Algunes d'aquestes aplicacions requereixen codificadors de gran presició amb una mida moderada. En aquesta situació són particularment útils els codificadors absoluts uni-corona. El major inconvenient que presenten aquest tipus de codificadors és la dificultat del disseny de codificadors de resolució arbitrària. L'objectiu d'aquest treball és obtenir un mètode per resoldre aquest problema de manera eficient.

La intenció és resoldre el problema usant eines matemàtiques. Per això caldrà formular el problema en el llenguatge de les matemàtiques i estudiar-ne les propietats. En el cas que el problema sigui massa difícil de resoldre amb les eines disponibles, caldrà elaborar-ne de noves o simplificar el problema a resoldre. Nosaltres farem les dues coses. Un cop disposem de les eines adequades podrem usar-les per construir una solució al problema que, en el nostre cas, prendrà la forma d'un algorisme. La motivació per usar mètodes matemàtics per resoldre el problema prové de l'article Ventura (1997). En aquest article l'autor utilitza mètodes algebraics per estudiar un problema semblant al que apareix quan es formula en llenguatge matemàtic el problema de la construcció de codificadors angulars absoluts.

A part de la solució del problema en sí, el projecte persegueix dos objectius secundaris. D'una banda, l'exposició clara dels mètodes matemàtics emprats per a la solució del problema. Aquesta exposició ha de permetre comprendre la relació entre els diferents objectes matemàtics considerats i la seva rellevància en la solució del problema. Per altra banda, identificar altres problemes de naturalesa enginyeril, que, a part de la codificació angular, es

puguin beneficiar dels resultats obtinguts. A més, es poden identificar possibles millores i línies de treball futur que es poden seguir a partir d'aquests resultats.

A continuació comentarem l'estructura de la memòria. Però abans ens agradaria remarcar que alguns dels resultats que s'exposen en els Capítols 2 i 3 d'aquesta memòria apareixeran publicats en un article de revista que ja ha estat acceptat (Fuertes *et al.*, 2008).

En el Capítol 1 introduïm el problema de la codificació i abordem la construcció de dispositius que permetin resoldre aquest problema. Per fer-ho, en primer lloc, plantejem quin és problema de la codificació angular i analitzem quines són les característiques que demanem a un dispositiu que el resolgui. A continuació, fem un recorregut per les diferents formes de resoldre el problema en ordre creixent de complexitat conceptual. Començant pels codificadors incrementals, passant pels codificadors absoluts multi-corona i acabant amb els codificadors absoluts uni-corona, comentem quines són les febleses i les qualitats de cadascun d'aquests mètodes. En relació als codificadors absoluts uni-corona, presentem els circuits LFSR i expliquem perquè les seqüències binàries que aquests circuits generen són útils en el disseny de codificadors angulars. La generalització d'aquest mètode és l'objectiu d'aquest treball.

El Capítol 2 presenta el llenguatge i els resultats matemàtics necessaris per abordar el problema. El concepte de (m, n, e)-seqüència es presenta com la generalització adequada de les seqüències generades per circuits LFSR. En termes de la codificació angular, es planteja el problema a resoldre com la construcció d'aquestes seqüències amb uns paràmetres donats. Un cop definit aquest objecte i plantejat el problema a resoldre, estudiem les seves propietats i justifiquem que la seva construcció mitjançant un algorisme de cerca per força bruta no és eficient. Això ens porta a considerar la relació d'aquests objectes amb els grafs dirigits de deBruijn. A partir d'aquí veiem que restringint-nos a uns valors particulars de m podem aplicar la teoria dels cossos finits i l'àlgebra lineal per trobar solucions al problema inicial. L'estudi de les propietats algebraiques d'aquestes solucions ocupa la resta del capítol i ens dóna eines per a la construcció d'un algorisme eficient per resoldre el problema.

Els mètodes matemàtics desenvolupats es posen en pràctica al Capítol 3 per construir l'esmentat algorisme. Usant els resultats del capítol anterior resolem en detall un exemple particular. Després, prenent l'exemple com a

inspiració, caracteritzem les propietats de la solució en el cas general. Aquesta caracterització ens permet reduir el domini de cerca i construir un algorisme per resoldre el problema. Un cop presentat l'algorisme, n'analitzem la complexitat i justifiquem la seva eficiència. Finalment estudiem mitjançant un experiment les propietats de les solucions que s'obtenen amb l'algorisme presentat.

La memòria acaba amb el Capítol 4. En aquest capítol presentem una altra aplicació de l'algorisme que hem obtingut. Concretament en el posicionament de vehicles guiats automàticament. Aquests vehicles poden usar seqüències com les que hem considerat per conèixer quina és la seva posició al llarg d'un recorregut que efectuen de manera automàtica. Els paràmetres típics d'aquest tipus d'aplicacions fan que el nostre algorisme resulti particularment útil. Per acabar el capítol, fem una sèrie de reflexions sobre els resultats obtinguts i les línies de treball que s'obren en vistes d'aquests.

Capítol 1

Introducció i motivació: la codificació angular

En aquest capítol plantejarem quin és el problema que desitgem resoldre i que motivarà la teoria que desenvoluparem en els capítols següents. Concretament voldrem ésser capaços de dissenyar codificadors angulars uni-corona de resolució arbitraria.

Per plantejar aquest problema, primer explicarem en què consisteix el problema de la codificació angular i què és un codificador angular. Un cop vist això, passarem a analitzar les diferents classes de codificadors angulars existents. Presentarem els codificadors incrementals i analitzarem com es comporten enfront d'errors. El caràcter acumulatiu d'aquests ens portarà a una altra família de codificadors que no sufreix aquest problema, els codificadors absoluts.

Estudiarem els codificadors absoluts multi-corona i justificarem la millora que suposa en termes d'espai l'ús de codificadors uni-corona. Un cop vist això presentarem el mètode estàndard de disseny de codificadors angulars absoluts uni-corona basat en seqüències generades mitjançant circuits LFSR. Acabarem senyalant les restriccions d'aquest mètode que justifiquen perquè es convenient disposar de mètodes pel disseny de codificadors uni-corona de resolució arbitraria.

1.1 El problema de la codificació angular

El problema de la codificació angular consisteix en mesurar amb una certa precisió la posició angular d'un eix rotatori respecte una referència mitjançant un dispositiu electromecànic que anomenarem codificador angular. Vegeu la Figura 1.1. La sortida d'aquest dispositiu ha de consistir en un senyal elèctric que representi d'alguna manera l'angle θ que es desitja mesurar. Per fixar idees, usarem radians per mesurar els angles i per tant tindrem $\theta \in [0, 2\pi)$. Tot i que en principi aquest senyal podria ésser de naturalesa analògica o digital, ens centrarem en codificadors angulars digitals doncs aquests són els més emprats.

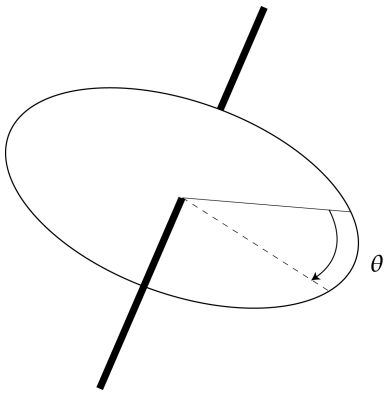

Figura 1.1: Volem mesurar el desplaçament angular θ de l'eix.

En la definició de codificació angular hem usat l'expressió "amb una certa precisió". Cap sistema electrònic pot representar la infinitat de nombres que hi ha en l'interval $[0, 2\pi)$, només se'n poden representar una quantitat finita. Aquesta és una de les raons que ens força a considerar les nostres mesures com a aproximacions al valor real de l'angle i que gaudeixen d'una certa precisió. En els codificadors angulars s'acostuma a prendre com a conjunt de possibles resultats de la mesura un conjunt de valors equiespaiats sobre l'interval $[0, 2\pi)$.

Per fer això cal dividir l'interval $[0, 2\pi)$ en un nombre enter e de subintervals disjunts consecutius de longitud $\Delta\theta = \frac{2\pi}{e}$. En aquestes condicions una mesura es correspon amb un número enter $0 \leq l \leq e - 1$, que indica en quin dels e intervals ens trobem situats. A partir d'aquest número l podem obtenir una aproximació de la posició angular real de l'eix amb una precisió

que dependrà de e. Denotem per $\hat{\theta} \in [0, 2\pi)$ l'aproximació de θ que podem obtenir a partir de l i la definim com

$$\hat{\theta} = l\Delta\theta + \frac{\Delta\theta}{2}.$$

Prenent el punt mig de l'interval on ens trobem pretenem minimitzar l'error comés en l'aproximació. Com més petit sigui l'error comés en aquesta aprocimació, més gran serà la precisió del nostre codificador. Si suposem que no hi ha hagut error en la lectura de l i que per tant la posició angular real de l'eix es troba en el subinterval l-èsim de $[0, 2\pi)$, podem assegurar que $\theta \in [l\Delta\theta, (l+1)\,\Delta\theta)$. D'aquí obtenim les següents cotes per l'error comés en la mesura de la posició angular de l'eix:

$$0 \leq \left|\theta - \hat{\theta}\right| \leq \frac{\Delta\theta}{2} = \frac{\pi}{e}.$$

Concloem doncs que l'error màxim comés pel codificador és inversament proporcional al nombre de particions de l'interval. O sigui, per disminuir l'error comés en les mesures cal augmentar el nombre de particions de l'interval. Com és habitual en enginyeria, se'ns planteja un compromís. Òbviament desitgem que el nostre codificador sigui capaç de mesurar posicions angulars amb un error molt petit. Per fer-ho cal que el cercle estigui dividit en un gran nombre d'arcs consecutius. Quan la mida del codificador està restringit per l'aplicació, la única solució és disminuir la mida físic dels arcs. Aquesta reducció en la mida dels arcs requereix de tecnologia més petita i precisa i, conseqüentment, més cara. Ens trobem davant el clàssic compromís entre cost i precisió. La decisió en cada cas dependrà de l'aplicació. Remarquem que el rang d'aplicacions dels codificadors és molt extens i variat, d'entre els quals, un dels més importants és la robòtica (Niku, 2001).

A continuació presentarem els diferents principis en què es basen els codificadors angulars més emprats i comentarem els avantatges i desavantatges de cadascun. Una classificació més extensa dels diferents tipus de codificadors angulars existents es pot trobar a Mayer (1999).

1.2 Codificació incremental

Segons la naturalesa de la referència segons la qual es mesura la posició, es poden distingir dues classes de codificadors angulars: els codificadors incre-

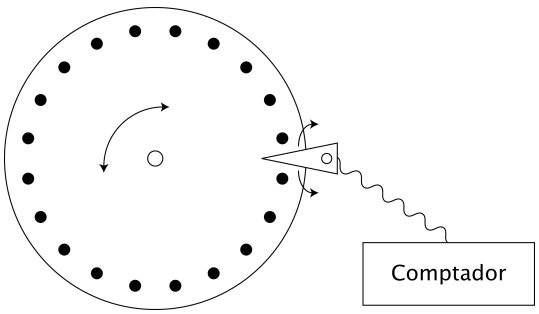

Figura 1.2: Diagrama esquemàtic d'un codificador incremental.

mentals i els codificadors absoluts. En aquesta secció presentem els codificadors incrementals.

La codificació incremental es caracteritza per obtenir la posició angular del sistema respecte la posició inicial d'aquest. Bàsicament consisteix en comptar quantes posicions ens hem allunyat de la posició inicial. Per fer-ho utilitzem un comptador inicialitzat a zero i prenem com a referència la posició on es troba l'eix en el moment de posar en marxa el sistema. A partir d'aquí anotem al comptador el nombre d'intervals que ens hem allunyat de la posició inicial. Òbviament, aquest comptatge és mòdul e doncs si ens allunyem e posicions de la posició inicial, en realitat tornem a la posició inicial.

Físicament, es dividiex la circumferència en e intervals iguals i es colloca una marca a cada canvi d'interval. Juntament amb l'eix es monta un dispositiu capaç de detectar les marques quan passen i en quin dels dos possibles sentits passen. Aquest dispositiu es conecta al comptador de manera que cada cop que detecta el pas d'una marca el comptador s'incrementa o es decrementa una unitat en funció de quin sigui el sentit de gir de l'eix. D'aquesta manera el comptador ens permet saber quina és la posició actual de l'eix respecte la posició de referència, és a dir, la inicial. A la Figura 1.2 es pot veure un esquema d'aquest muntatge.

El cost de fabricació d'aquest tipus de codificadors és baix degut a la simplicitat del seu funcionament. No obstant també presenten un problema important: poca robustesa enfront d'errors. Concretament, els errors són de caràcter acumulatiu. Vegem-ne un exemple. Suposem que el comptador conté el valor l i fins aquest moment no ha ocurregut cap error. Això significa

que l'error en aquest moment és

$$\left|\theta_0 - \hat{\theta}_0\right| \leq \frac{\Delta\theta}{2} = \frac{\pi}{e}$$

on θ_0 és el valor de la posició angular i $\hat{\theta}_0$ l'estimació. Ara suposem que l'eix gira en sentit positiu i rebassa una marca però el detector no ho detecta; s'ha produit un error. El nou valor de la posició angular és θ_1 i compleix $\hat{\theta}_0 + \frac{\Delta\theta}{2} \leq \theta_1 < \hat{\theta}_0 + \frac{3\Delta\theta}{2}$. En canvi, com que el valor del comptador no s'ha actualitzat, el valor de l'estimació continua essent el mateix: $\hat{\theta}_1 = \hat{\theta}_0$. Això ens permet donar una cota superior i una cota inferior per a l'error de mesura en aquesta nova posició:

$$\frac{\Delta\theta}{2} \leq \left|\theta_1 - \hat{\theta}_1\right| \leq \frac{3\Delta\theta}{2}.$$

Aquesta cota superior és substancialment major que la de l'error en la posició anterior. I la cota inferior indica que la lectura té error segur. Per comprovar el caràcter acumulatiu dels errors suposem que ara que l'eix es torna a desplaçar en el mateix sentit que abans i que rebassa una nova marca que tampoc és detectada. Si denotem la posició angular real per θ_2 i l'estimació, que continua essent igual que abans, $\hat{\theta}_2 = \hat{\theta}_0$, un anàlisi similar a l'anterior ens donarà com a resultat les cotes

$$\frac{3\Delta\theta}{2} \leq \left|\theta_2 - \hat{\theta}_2\right| \leq \frac{5\Delta\theta}{2}.$$

O sigui, l'error total creix a mesura que els errors es succeeixen.

Notem el següent: en la situació plantejada, un nou error de detecció que es produís en el sentit de gir oposat als que s'han produït fins ara es cancel·laria amb un dels anteriors. Aquest fet faria disminuir l'error total de la mesura. No obstant aquest no serà sempre el cas. És per aquesta raó que els codificadors incrementals no s'usen en sistemes on els errors són crítics o en sistemes de difícil reparació.

A la pràctica, codificadors d'aquest estil s'usen per mesurar velocitats angulars enlloc de posicions. Un exemple són els sistemes ABS que incorporen alguns vehicles com el que es pot veure a la Figura 1.3.

Figura 1.3: Sistema ABS a la roda delantera d'una motocicleta.

1.3 Codificació absoluta

La codificació absoluta permet mesurar la posició angular de l'eix respecte una referència fixa. I si bé és cert que els codificadors incrementals poden oferir mesures absolutes respecte una referència fixa, implementant un protocol de retorn a una posició de repós coneguda cada cop que s'inicialitza el sistema, en aquesta secció considerarem uns codificadors basats en uns altres principis i que són absoluts per naturalesa. D'entrada podem dir que la codificació absoluta resol el problema de l'acumulació d'errors que presenta la codificació incremental.

Igualment com en la codificació incremental, en la codificació absoluta es divideix la circumferència en e intervals iguals. Però aquesta vegada marquem cada interval amb una marca diferent i equipem l'eix amb un dispositiu que anomenarem lector capaç de llegir la marca corresponent a l'interval situat en la posició de referència. Vegeu la Figura 1.4. D'aquesta manera, llegint la marca i mirant en una taula a quina posició correspon sobre la circumferència podem saber el desplaçament angular de l'eix respecte l'origen.

Els components fonamentals d'un codificador absolut són doncs, les diferents marques i el lector que permet llegir-les. El disseny d'unes "bones" marques serà l'objectiu dels pròxims capítols. De moment ens centrarem a exposar les alternatives existents més corrents.

La primera pregunta que un es pot fer davant d'aquest repte de disseny és quina de les dues vies següents és més pràctica: escollir e marques diferents i després dissenyar un lector capaç de distingir-les; o bé escollir un lector

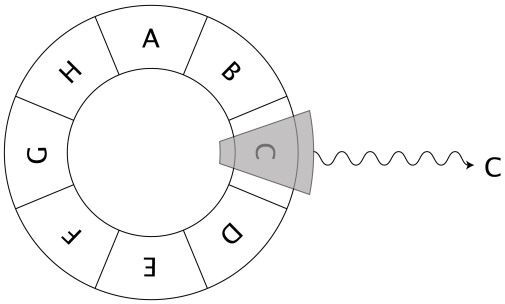

Figura 1.4: Diagrama esquemàtic d'un codificador absolut.

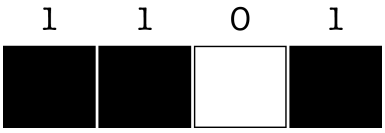

Figura 1.5: Marques binàries que representen el nombre 13.

capaç de distingir entre moltes marques i després escollirne e d'entre totes les possibles? Com a enginyers, òbviament escollirem la segona. Aquesta respon als principis d'escalabilitat, permetent usar el mateix lector per obtenir codificadors de diferents resolucions; i de reaprofitament d'un lector ja existent, reduint el temps i la complexitat del disseny del codificador.

El lector emprat en la majoria de codificadors absoluts consisteix en un lector de nombres binaris. Aquests venen representats per dos tipus de marques diferents que representen uns i zeros. A la Figura 1.5 podem veure una d'aquestes marques que representa el nombre 13 en binari amb el bit de més pes a l'esquerra. En aquests cas una taca negra significa que el bit val "1" i una de blanca que el bit val "0". Físicament aquests lectors estan formats per tants detectors com bits són capaços de llegir. Cada detector es situa sobre una de les taques que conformen la marca i detecta si es tracta d'un "1" o d'un "0". El tipus de detector ha d'anar d'acord amb la naturalesa física de les marques. Dos dels sistemes més utilitzats són, d'una banda, els detector òptics que permeten distingir marques de color blanc o negre, o forat i no-forat; i de l'altra, els dectectors elèctrics que permeten distingir marques formades per zones conductores i no-conductores del corrent.

Un cop considerats els lectors i escollit un lector de nombres binaris, procedirem a considerar les marques.

Disposant d'un lector de nombres binaris, el primer que se'ns acudeix és etiquetar els sectors de forma consecutiva començant pel primer, tal com es mostra a la Figura 1.6. En aquest cas el disc queda dividit en diverses corones concèntriques i el sensor que llegeix la marca està compost de tants detectors com corones. Observem que el nombre de corones vé donat pel nombre de bits necessaris per representar el número més gran, és a dir $\lceil \log_2 e \rceil$ si numerem les posicions $0, 1, \ldots, e-1$.

Una variant d'aquest primer mètode és la que s'usa àmpliament en la industria avui en dia. Consisteix en prendre un nombre d'intervals de la forma $e = 2^k$ i marcar els e intervals amb els nombres $0, 1, \ldots, 2^k - 1$ en binari, però usant l'ordre donat per un codi de Gray enlloc de l'ordre natural. La Figura 1.7 n'és un exemple amb $e = 8$. Observem que en aquest cas el nombre de corones és $k = 3$. En una codificació de Gray només hi ha un bit de diferència entre dues posicions consecutives qualssevulla. Això ofereix un avantatge substancial respecte la codificació natural en el següent context: suposem que l'eix està girant i just en el moment de transició d'un interval a un altre consultem el valor del sensor. L'estat dels detectors pot reflexar tant el valor de l'interval anterior com el del següent, no està clar. No obstant, com que entre un interval i el següent només canvia un dels bits, la incertesa es centra només en un únic bit. I per tant l'error queda limitat a un interval com a màxim. En el cas de la codificació natural podem tenir els detectors en una barreja entre els dos intervals consecutius que ens acabi donant el valor d'una posició que no es correspongui amb cap dels dos i que inclús caigui lluny dels dos. Observem que la codificació incremental no presenta aquest problema.

El problema principal dels dos sistemes de codificació absoluta presentats fins ara el trobem quan considerem resolucions elevades. El nombre de corones, i per tant la mida del codificador, creix amb la resolució. És per aquesta raó que s'aborda l'estudi de les codificacions absolutes uni-corona en contraposició a les codificacions multi-corona considerades fins ara. Però abans de prendre aquesta direcció farem un estudi quantitatiu per justificar que la reducció de múltiples corones a una sola corona redueix substancialment la mida del codificador quan considerem resolucions elevades.

Suposem que volem un codificador multi-corona de resolució e. El nombre

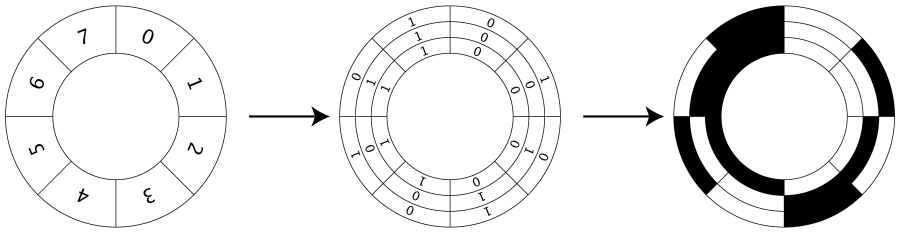

Figura 1.6: Codificador absolut de 3 bits codificat en binari natural.

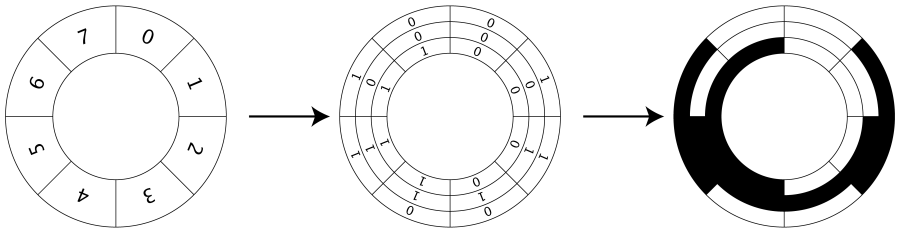

Figura 1.7: Codificador absolut de 3 bits codificat en codi Gray.

de corones necessàries serà

$$k = \lceil \log_2 e \rceil.$$

Denotarem per r i R els radis interior i exterior de la corona completa. Suposarem que els detectors de què disposem requereixen que la marca ocupi, com a mínim, un sector de corona circular com el de la Figura 1.8. Això ens dóna les mides mínimes de la corona completa

$$r = \frac{ae}{2\pi} \qquad \text{i} \qquad R = kb + r.$$

Observem que si uséssim una sola corona el radi exterior seria

$$R' = b + r$$

Usant aquestes dades podem calcular les arees A i A' ocupades, respectivament, pel codificador multi-corona i pel codificador uni-corona. El quocient d'aquestes dues arees ens dóna una idea de l'estalvi en espai al passar d'un codificador multi-corona a un d'uni-corona.

$$\frac{A}{A'} = \frac{\pi(R^2 - r^2)}{\pi(R'^2 - r^2)} = \frac{(R - r)(R + r)}{(R' - r)(R' + r)} = k\left(\frac{kb + 2r}{b + 2r}\right)$$

Suposant que a i b són del mateix ordre de magnitud i que e és suficientment gran, obtenim que aproximadament

$$\frac{A}{A'} \approx k \approx \log_2 e.$$

Si considerem, per exemple, un codificador amb resolució d'una dècima de grau, és a dir, $e = 3600$, obtenim que el codificador multi-corona ocupa unes 12 vegades més àrea que l'uni-corona.

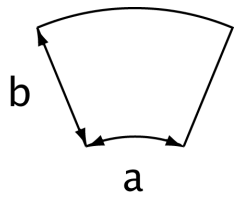

Figura 1.8: Sector de corona circular.

1.4 Codificació absoluta uni-corona

Hem vist que la principal motivació per a l'ús de codificadors absoluts uni-corona és la reducció en l'espai ocupat pel codificador. Ara descriurem com es construeixen els codificadors d'una sola corona més coneguts. Aquests seran els que generalitzarem en els pròxims capítols.

Matenint la restricció d'usar detectors de nombres binaris, si volem un codificador de resolució e necessitarem llegir almenys $\lceil \log_2 e \rceil$ bits. Si els volem llegir d'una sola corona, necessitarem aquest nombre de detectors sobre la corona. I si volem que els detectors llegeixin bits diferents, cada detector haurà d'apuntar a una posició diferent de la corona. La primera manera que se'ns acudeix és llegir posicions consecutives sobre la corona. A les Figures 1.10 i 1.11 es poden apreciar les diferències en la disposició dels detectors en un codificador multi-corona i en un uni-corona com el descrit per a $e = 8$.

El primer i més conegut codificador angular absolut uni-corona que utilitza aquesta distribució dels detectors és el basat en seqüències binaries pseudo-aleatòries i apareix per primer cop a Good (1946). Aquestes seqüències es generen mitjançant registres de desplaçament linealment realimentats, o LFSR. El nombre de bits d'aquests registres s'anomena la mida del LFSR. A la Figura 1.9 es pot veure un LFSR de mida 4.

El funcionament és el següent. Si el LFSR té mida k, s'inicialitzen els registres $u_0, \ldots, u_{k-1}$ amb un bit "1" o "0", però de manera que no siguin tots "0". A cada tic del rellotge es produeix l'actualització $u_i \leftarrow u_{i+1}$ per a $i = 0, \ldots, k-2$. El valor de u_{k-1} es calcula utilitzant el valor dels registres i

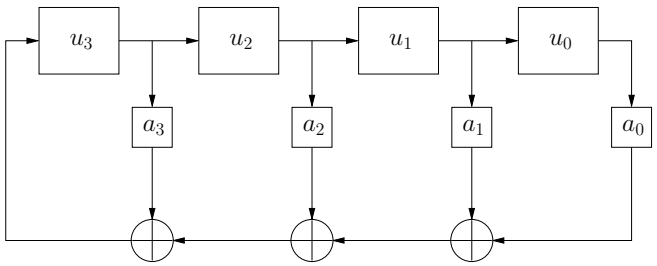

Figura 1.9: LFSR de mida $k = 4$.

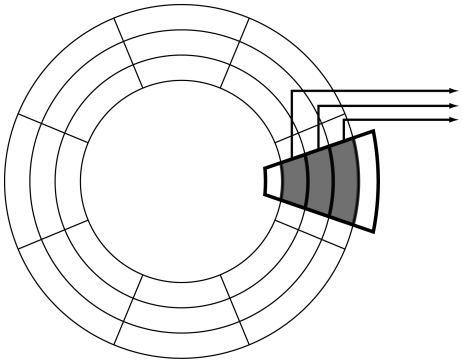

Figura 1.10: Disposició dels detectors en un codificador multi-corona.

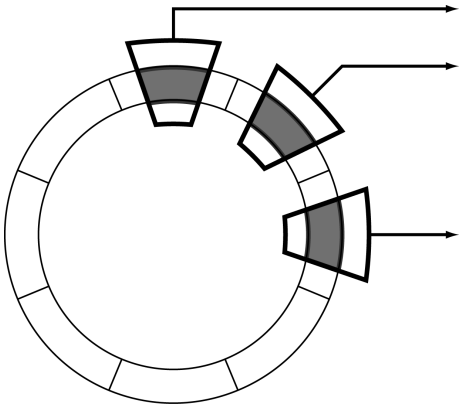

Figura 1.11: Disposició dels detectors en un codificador uni-corona.

els coeficients (també binaris) $a_0, \ldots, a_{k-1}$ segons

$$u_{k-1} \leftarrow a_0 u_0 + \cdots + a_{k-1} u_{k-1}.$$

Observem que efectivament la realimentació que actualitza u_{k-1} és una funció lineal del valor de tots els registres. La seqüència formada pels valors que pren el registre u_0 en cada tic de rellotge és la seqüència generada pel LFSR de mida k amb connexions $a_0, \ldots, a_{k-1}$ i configuració inicial $u_0, \ldots, u_{k-1}$.

Òbviament, si la configuració inicial és $u_0 = \cdots = u_{k-1} = 0$, aleshores el valor de tots els registres sempre es manté a zero i la seqüència generada està formada per tot zeros. Per altra banda, es pot veure que sota unes condicions adecuades pels coeficients $a_0, \ldots, a_{k-1}$ la seqüència generada és periòdica i maximal, en el sentit que mirant els registres com a vectors $(u_0, \ldots, u_{k-1})$ passem per tots els $2^k - 1$ valors possibles llevat del $(0, \ldots, 0)$. Aquestes seqüències s'anomenen pseudo-aleatòries ja que compleixen els postulats d'aleatorietat de Golomb, vegeu Golomb (1981).

Prenent un període sencer d'una seqüència pseudo-aleatòria generada per un LFSR de mida k podem obtenir un codificador absolut uni-corona de resolució $2^k - 1$ usant k detectors. Òbviament, si el període de la seqüència és $2^k - 1$ tenim aquest nombre de bits per col·locar a la corona circular. A més, com que el LFSR passa per tots els vectors $(u_0, \ldots, u_{k-1})$ possibles llevat del nul abans de repetir-se, cadascuna de les paraules de k bits consecutius que es pot llegir sobre la corona circular és diferent i per tant identifica de manera única una posició del codificador angular. Vegem-ne un exemple.

Prenem un LFSR de mida $k = 3$ amb connexions $a_0 = a_2 = 1$ i $a_1 = 0$. Si prenem configuració inicial $u_0 = u_1 = 0$ i $u_2 = 1$ aleshores la seqüència d'estats del LFSR en la forma $u_0 u_1 u_2$ és la següent:

$$001, 011, 111, 110, 101, 010, 100, 001, 011, \ldots$$

Hem pres les connexions de manera que la seqüència generada és maximal. Per tant, com és d'esperar, la seqüència d'estats té període $2^3 - 1 = 7$. La seqüència generada pel LFSR formada pel bit de més a l'esquerra de cadascun dels estats de la seqüència és

$$00111010011101001110 1 \ldots$$

Prenent un període de la seqüència en qüestió, 0011101, podem construir un

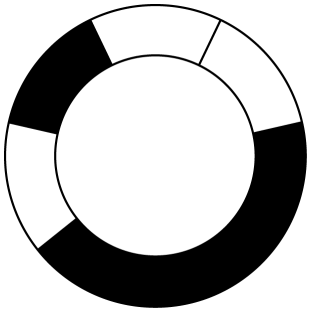

Figura 1.12: Codificador uni-corona basat en una seqüència pseudo-aleatòria.

codificador uni-corona amb $e = 7$ i 3 detectors com el de la Figura 1.12. Usant la seqüència 00011101 que s'obté d'afegir un zero al principi de la seqüència generada pel LFSR es pot construir un codificador amb $e = 8$ i 3 detectors.

Aquest mètode permet obtenir codificadors angulars absoluts uni-corona d'una manera senzilla ja que les conexions necessàries per obtenir LFSRs amb períodes maximals estan ben estudiades. No obstant, aquest mètode no permet construir codificadors de resolució arbitrària. Les possibles resolucions han de ser de la forma $2^k - 1$ per algun $k \geq 2$, o 2^k si afegim artificialment un 0 de manera que la seqüència contingui tots els 2^k vectors binaris possibles de longitud k. El treball recollit en els dos pròxims capítols permet generalitzar aquesta aproximació a resolucions arbitràries pagant un preu: assumir la possibilitat d'usar més detectors dels que serien necessaris en un codificador multi-corona de la mateixa resolució. A més, el mètode ens permetrà dissenyar codificadors uni-corona de resolució arbitrària que utilitzin detectors de més de dos símbols.

El mètode emprat és essencialment matemàtic i descriu un procediment eficient per obtenir seqüències útils per a la construcció de codificadors angulars absoluts uni-corona. No obstant, degut a l'interès de les possibles aplicacions d'aquest tipus de codificadors, existeixen altres mètodes basats en procediments empírics que permeten usar aquestes seqüències en la construcció de codificadors de resolució arbitrària. La majoria d'aquests mètodes es basa en el truncament de seqüències maximals i l'ús d'una circuiteria complexa per a la recuperació de la posició real a partir de la lectura dels detectors. Vegeu, per exemple, Petriu (1985).

Capítol 2

Modelització matemàtica

En aquest capítol presentem les eines matemàtiques necessàries pel disseny de codificadors angulars absoluts uni-corona. En primer lloc definim un concepte combinatori, el de m-seqüència, que recull i generalitza les propietats que fan tan interessants les seqüències generades mitjançant circuits LFSR pel disseny de codificadors angulars. Això ens permetrà formular el problema matemàtic subjacent que cal resoldre pel disseny d'un codificador angular amb uns paràmetres donats. Un cop formulat aquest problema justificarem que no és possible resoldre'l eficientment de manera algorísmica mitjançant la força bruta.

A través d'un seguit de reduccions que ens portaran de les m-seqüències als grafs de deBruijn, després a l'àlgebra lineal sobre cossos finits i finalment als polinomis sobre cossos finits, transformarem el problema combinatori inicial en un problema de naturalesa algebraica.

Estudiant aquest problema algebraic desenvoluparem una teoria que ens permetrà concloure que amb els nostres mètodes no sempre és possible obtenir una solució exacte pel problema inicial. Un cop vist això, plantejarem un nou problema que ens permetrà donar una solució aproximada al problema inicial de manera que l'aproximació minimitzi, dins les possibilitats del nostre mètode, el nombre de detectors emprats en la construcció del codificador angular i, conseqüentment, el cost d'aquest codificador.

La teoria desenvolupada en aquest capítol ens servirà de base per, en el capítol següent, donar un algorisme que permeti obtenir la millor aproximació possible a la solució del problema inicial mitjançant aquests mètodes.

2.1 m-seqüències tancades

Com ja hem vist en 1.4 els LFSR es poden usar per generar seqüències binàries (formades pels símbols "1" i "0") amb unes propietats particularment útils pel disseny de codificadors angulars absoluts uni-corona. Ara pretenem abstreure les característiques que fan aquestes seqüències tant interessants pels nostres propòsits i recollir-les en una definició formal que sigui independent del mètode usat per generar la seqüència. Aquest concepte serà el de m-seqüència tancada.

Per començar, recordem les propietats de les seqüències generades pels LFSR.

1. Seqüències de longitud $2^n - 1$ amb n un nombre natural.
2. Considerant la seqüència com a periòdica, totes les $2^n - 1$ paraules formades per n símbols consecutius són diferents.
3. Propietats estadístiques de seqüència pseudo-aleatoria.

Al fer ús d'aquestes seqüències per marcar la corona circular d'un codificador angular, essencialment estem interessats en la propietat 2. La propietat 1 representa una restricció: no podem escollir arbitràriament la resolució del nostre codificador sinó que ens hem de conformar amb una de la forma $2^n - 1$. En aquesta aplicació, la propietat 3 es mostra pràcticament irrellevant. A part d'aquestes propietats, notem que el fet de generar-les usant circuits digitals implica que són seqüències sobre l'alfabet $\{0, 1\}$.

A partir de totes aquestes observacions sobre les seqüències generades pels LFSR, és natural plantejar-se la següent definició com una generalització de tals seqüències (com a mínim, en el nostre context d'aplicació).

Definició 2.1. Una (m, n, e)-*seqüència tancada* és una paraula de longitud e sobre un alfabet de m símbols de manera que les e subparaules formades per n símbols circularment consecutius són diferents dos a dos. Segons l'èmfasi que volguem fer en els paràmetres, també parlarem de m-seqüències tancades o simplement de seqüències tancades.

Considerarem que dues tals seqüències són la mateixa si una és igual a l'altra llevat d'una permutació circular. En aquest cas, el conjunt de subparaules de les dues seqüències és el mateix.

Hem apuntat que aquesta definició generalitza les seqüències generades pels LFSR. Els següents dos exemples pretenen aclarir aquest punt.

Exemple 2.1. Els LFSR generen $(2, n, 2^n - 1)$-seqüències tancades.

Exemple 2.2. Un exemple de $(3, 2, 9)$-seqüència tancada és (001122021). Les 9 subparaules de longitud 2 són: 00, 01, 11, 12, 22, 20, 02, 21 i 10. Tal com hem explicat, considerem que aquesta seqüència i qualsevol permutació circular seva, per exemple la seqüència (122021001), són la mateixa.

Aquesta seqüència no encaixa en la catergoria de les generades pels LFSR per dues raons: utilitza 3 símbols diferents i la seva longitud total no és de la forma $2^n - 1$.

Aquest exemple es pot usar per contruir un codificador angular absolut de 9 posicions si disposem de detectors que distingeixin tres símbols diferents. En general, una (m, n, e)-seqüència tancada ens permet construir un codificador angular absolut uni-corona de resolució e usant n detectors capaços de distingir m símbols.

Sobre un alfabet de m símbols hi ha exactament m^n paraules diferents de longitud n. Així doncs, de la definició de (m, n, e)-seqüència tancada es desprén directament la restricció $e \leq m^n$ sobre els paràmetres. Una pregunta natural en aquest punt és: existeixen més restriccions? Per a quins valors de (m, n, e) existeixen aquestes seqüències? La resposta és que no existeixen més restriccions i ens la dóna el següent teorema.

Teorema 2.2. *Siguin m, n i e enters positius tals que $e \leq m^n$. Llavors existeix una (m, n, e)-seqüència tancada.*

Una demostració d'aquest resultat en termes combinatoris es pot trobar a Lempel (1971). Malauradament, la demostració ens assegura l'existència de la seqüència però no ens dóna cap idea sobre com construir-la de manera eficient.

Així doncs, sabem que existeixen (m, n, e)-seqüències tancades que poden ser usades per construir codificadors angulars de qualsevol resolució. El problema que ara se'ns planteja és, donats m, n i e satisfent les restriccions del teorema, com construir una tal seqüència.

Abans d'abordar el problema de la construcció d'aquestes seqüències convé fer una parada i estudiar la rellevància per a la nostra aplicació dels tres paràmetres que caracteritzen una seqüència. Per analogia amb el cas de les seqüències generades pels LFSR tenim que e serà la resolució del codificador i m el nombre de símbols diferents que els detectors emprats són capaços de distingir. A més, n es correspon amb el nombre de detectors necessaris

per construir el codificador. Tal com hem comentat en el Capítol 1, la longitud e vindrà donada com una especificació de disseny. El paràmetre m representa una restricció tecnològica dels nostres detectors i el podem pensar també com un paràmetre de disseny. Per a n tenim la restricció ja comentada de $n \geq \lceil \log_m e \rceil$. Segons el Teorema 2.2 sempre existeix una solució amb $n = \lceil \log_m e \rceil$. No obstant, com veurem en la pròxima secció, trobar aquesta solució és, en general, un problema difícil des del punt de vista computacional. En el cas en què no ens sigui possible trobar una seqüència amb $n = \lceil \log_m e \rceil$ exactament, buscarem una seqüència amb n el més petit possible. D'aquesta manera minimitzarem el nombre de detectors usats en la construcció del codificador i per tant el seu cost.

Com a conclusió d'aquesta secció enunciarem de manera formal el problema que ens agradaria resoldre en un principi. Al llarg d'aquest capítol anirem transformant i simplificant el problema fins arribar a un de més fàcil de resoldre la solució del qual ens doni una bona aproximació a la solució d'aquest primer problema.

Problema 1. Donats dos enters positius m i e, trobar de manera eficient una (m, n, e)-seqüència tancada amb $n = \lceil \log_m e \rceil$.

2.2 La inviabilitat de la força bruta

Per tal de resoldre el Problema 1, i inspirats per la seva naturalesa combinatòria, la primera solució que se'ns acudeix és la més senzilla conceptualment: explorem totes les possibles paraules de longitud e sobre l'alfabet de m símbols fins que en trobem una que es correspongui amb una (m, n, e)-seqüència tancada. En algorísmia, aquesta estratègia rep el nom de "força bruta". A continuació justificarem que aquesta solució no és computacionalment factible per raons d'eficiència.

Sobre un alfabet de m símbols, existeixen m^e paraules diferents de longitud e. Si considerem que dues paraules són la mateixa quan una és una permutació circular de l'altra, com que donada una paraula genèrica podem fer e permutacions circulars diferents, el nombre total de paraules essencialment diferents és, aproximadament,

$$P(m, e) \approx \frac{m^e}{e}.$$

Aquesta fórmula no és exacte perquè per a les paraules que són periòdiques les e permutacions circulars no donen e paraules diferents, sinó que en donen un divisor de e. Per tant aquesta és una estimació a la baixa del nombre de paraules circularment diferents de longitud e sobre un alfabet de m símbols, però prou bona pels nostres propòsits. I d'aquestes, quantes són (m, n, e)-seqüències tancades? Actualment, desconeixem la resposta general a aquesta pregunta, però si que coneixem la resposta per a un cas particular. Aquest cas serà suficient per justificar l'ineficàcia d'aquesta aproximació al problema.

A Rosenfeld (2002) es demostra que el nombre de (m, n, m^n)-seqüències tancades vé donat per

$$S(m, n) = \frac{(m!)^{m^{n-1}}}{m^n}.$$

En aquest cas, la proporció entre nombre de seqüències i el nombre total de paraules diferents vindrà donada per l'expressió

$$R(m, n) = \frac{S(m, n)}{P(m, m^n)} \approx \frac{(m!)^{m^{n-1}}}{m^{m^n}} = \left(\frac{m!}{m^m}\right)^{m^{n-1}}.$$

Intuïtivament, aquesta proporció ens dóna la probabilitat d'obtenir una (m, n, m^n)-seqüència tancada si prenem a l'atzar (de manera uniforme) una seqüència de les $P(m, m^n)$ possibles. També podem pensar-ho com que $R(m, n)^{-1}$ representa, si ens posessim a explorar totes les possibles paraules una per una, el nombre de paraules que hauríem de provar de mitjana fins a trobar-ne una que fos una (m, n, m^n)-seqüència tancada.

Usant l'aproximació de Stirling pel factorial, $m! \approx m^m \mathrm{e}^{-m}\sqrt{2\pi m}$, veiem que la fracció que apareix a $R(m, n)$ es comporta asimpòticament com

$$\frac{m!}{m^m} \approx \frac{\sqrt{2\pi m}}{\mathrm{e}^m},$$

i per tant decreix cap a 0 de manera exponencial quan incrementem m. Tenint en compte l'exponent al qual es troba elevada la fracció, podem concloure que, amb n fixada, $R(m, n)$ decreix cap a 0 de forma doblement exponencial a mesura que incrementem m. Per altra banda, si fixem $m \geq 2$ tindrem que

$$\left(\frac{\sqrt{2\pi m}}{\mathrm{e}^m}\right)^m < 1$$

i per tant el valor de $R(m,n)$ amb m fixada també decreix exponencialment cap a zero amb n. Així doncs podem concloure que per a valors grans de m i n la densitat de (m,n,m^n)-seqüències tancades dins del conjunt de totes les paraules cícliques de longitud m^n és molt petita.

Numèricament, ho podem veure en uns quants exemples en les Taules 2.1 i 2.2. Cada fila correspon a un valor diferent de m i cada columna a un valor de n. Observem que per valors relativament petits de m i n (que corresponen a longituds de la seqüència moderats) l'ordre de magnitud de $R(m,n)$ és astronòmicament petit. Això prova que explorar directament l'espai de totes les seqüències fins a trobar-ne una d'adequada és computacionalment inviable.

A tall orientatiu, podem substituir $e = m^n$ en l'expressió de $R(m,n)$ i obtenir una expressió $R(m,e)$ que ens doni una idea de la dificultat de trobar una (m,n,e)-seqüència tancada per força bruta amb $n = \lceil \log_m e \rceil$.

$$R(m,e) \approx \left(\frac{m!}{m^m}\right)^{\frac{e}{m}}.$$

Amb l'objectiu de poder comparar l'eficiència de l'algorisme que presentarem en el Capítol 3 amb la cerca per força bruta, suposarem que la mitjana del nombre d'operacions que ha de realitzar la cerca per força bruta per tal de trobar una (m,n,e)-seqüència tancada amb $n = \lceil \log_m e \rceil$ és de l'ordre de $R(m,e)^{-1}$. És a dir, el cost esperat creix de manera exponencial en e. El cost també creix amb m, però en aquest cas no de manera exponencial.

Hem justificat en aquesta secció que el Problema 1 és difícil de resoldre per força bruta. Per tant es fa necessari buscar un mètode més astut que la força bruta per construir seqüències tancades. En la pròxima secció introduirem una caracterització de les seqüències tancades que ens permetrà reformular el Problema 1 en un altre llenguatge i ens encaminarà cap a una possible solució.

2.3 Cicles sobre digrafs de deBruijn

Fins ara hem vist que si $e \leq m^n$ aleshores sempre existeix una (m,n,e)-seqüència tancada però que en general no sabem com construir-la de manera eficient. La resta d'aquest capítol el dedicarem a resoldre aquest problema per a un conjunt de m's particulars.

	5	6	7
2	32	64	128
3	243	729	2187
4	1024	4096	16384

Taula 2.1: Longitud de la seqüència obtinguda

	5	6	7
2	10^{-5}	10^{-10}	10^{-20}
3	10^{-53}	10^{-159}	10^{-477}
4	10^{-264}	10^{-1053}	10^{-4211}

Taula 2.2: Ordre de magnitud de $R(m, n)$

Per tal d'atacar el problema, transformarem la qüestió purament combinatòria de buscar una seqüència de longitud e sobre un alfabet de m símbols que compleixi les condicions d'una (m, n, e)-seqüència tancada en una qüestió sobre la dinàmica d'uns certs sistemes algebraics. Veurem que hi ha uns sistemes algebraics que permeten obtenir seqüències de la forma desitjada. De l'estudi d'aquests sistemes obtindrem unes condicions que s'han de complir per tal que la seqüència s'ajusti als paràmetres desitjats.

Aquest mateix problema es pot plantejar i estudiar utilitzant diversos llenguatges matemàtics diferents. El mètode que adoptem aquí pretén assolir un compromís entre el nivell d'abstracció necessari per treballar amb rigor i el nivell empíric en qué es mou la intuició. Per tal de facilitar-ne la comprensió acompanyarem l'exposició amb exemples sempre que ho considerem necessari.

Començarem definint una família de grafs dirigits, o digrafs, que depén de dos paràmetres m i n, de manera que qualsevol (m, n, e) seqüència tancada es correspon amb un cicle de longitud e sobre el digraf i viceversa. Sigui M un alfabet de m símbols i considerem el producte cartesià $V_{mn} = M^n$ que conté totes les paraules de longitud n sobre M. És a dir, si $x \in V_{mn}$ llavors $x = (x_1, \ldots, x_n)$ amb $x_i \in M$ per a $1 \leq i \leq n$. Aquest conjunt seran els vèrtexs del nostre digraf.

Per definir els arcs del digraf introduïm la *relació de desplaçament* entre les paraules de V_{mn}. Si $x, y \in V_{mn}$, direm que y és un desplaçament de x i ho denotarem per $x \to y$ si es compleix:

$$(x_2, \ldots, x_n) = (y_1, \ldots, y_{n-1}).$$

Usant aquesta relació definim el conjunt d'arcs del nostre digraf, $E_{mn} \subset V_{mn} \times V_{mn}$, com:

$$E_{mn} = \{(x, y) \,|\, x \to y\}.$$

El digraf que té V_{mn} per conjunt de vèrtexs i E_{mn} per conjunt d'arcs s'anomena *digraf n-dimensional de deBruijn de m símbols* i es denota per $B(m, n)$. La Figura 2.1 en mostra un exemple.

A continuació farem evident la relació entre els cicles de longitud e de $B(m, n)$ i les (m, n, e)-seqüències tancades. Sigui $x = (x_0, x_1, \ldots, x_{e-1})$ una (m, n, e)-seqüència sobre un alfabet M, $x_i \in M$ per $0 \leq i \leq e - 1$. Denotem per x^j la subparaula $(x_j, x_{j+1}, \ldots, x_{j+n-1})$ de x on considerem els subíndexs mòdul e. Observem que $x^j \in V_{mn}$ per $0 \leq j \leq e-1$. Aquestes e paraules són

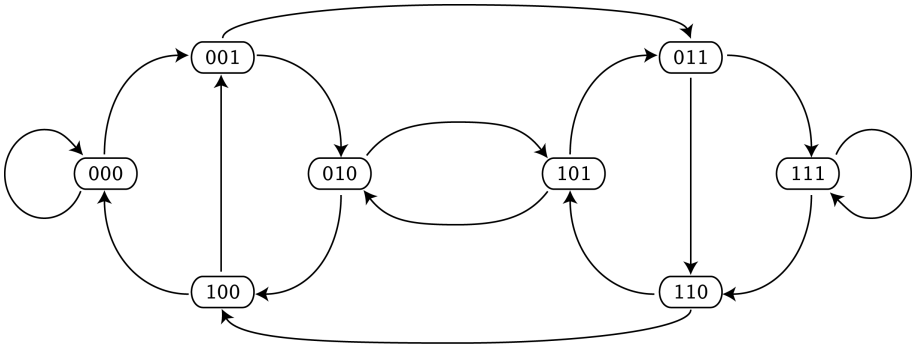

Figura 2.1: Digraf de deBruijn $B(2,3)$.

totes diferents per construcció doncs es corresponen amb les e subparaules de la (m,n,e)-seqüència tancada. A més, per construcció, tenim $x^j \to x^{j+1}$ per a tot $0 \leq j \leq e-2$ i $x^{e-1} \to x^0$. Per tant la seqüència de vèrtexs $x^0, x^1, \ldots, x^{e-1}, x^0$ de $B(m,n)$ forma un camí tancat sense repeticions, és a dir, un cicle. Igualment, qualsevol cicle de longitud e sobre el digraf $B(m,n)$ defineix una (m,n,e)-seqüència tancada.

Exemple 2.3. És fàcil veure que la seqüència de vèrtexs (001), (011), (110), (100), (000) forma un cicle de longitud 5 de $B(2,3)$ que està associat a la $(2,3,5)$-seqüència tancada (00110).

Mitjançant aquesta construcció podem traduïr l'enunciat combinatori del Teorema 2.2 al llenguatge de cicles i digrafs.

Teorema 2.3. *En el digraf de deBruijn $B(m,n)$ existeixen cicles de qualsevol longitud e tal que $1 \leq e \leq m^n$.*

A més, els digrafs de deBruijn permeten enunciar el següent problema, equivalent, com hem vist, al Problema 1.

Problema 2. Donats dos enters positius m i e, trobar de manera eficient un cicle de longitud e en el digraf $B(m,n)$, on $n = \lceil \log_m e \rceil$.

En la pròxima secció veurem com podem resoldre aquest problema usant permutacions.

2.4 Cicles induïts per aplicacions

Hem transformat el nostre problema inicial de trobar una (m, n, e)-seqüència tancada en el problema de trobar un cicle de longitud e sobre el digraf $B(m, n)$. En general, el problema de trobar cicles d'una determinada longitud sobre un digraf, tot i conèixer-ne l'existència, és difícil de tractar de manera algorísmica. No obstant això, aquest canvi de llenguatge ens aporta una nova perspectiva des d'on mirar-nos el problema. Començarem recordant un resultat clàssic de la Combinatòria.

Teorema 2.4. *Tota permutació d'un conjunt finit descomposa de manera única en un conjunt de cicles disjunts.*

Recordem que una permutació d'un conjunt no és res més que una bijecció del conjunt en ell mateix. Donats un conjunt finit X i una bijecció $F : X \to X$, definim l'*estructura cíclica* de F com el conjunt format per les longituds de tots els cicles que apareixen en la descomposició cíclica de F. Denotarem aquest conjunt mitjançant el símbol $\mathcal{CS}(F)$. Donat un element $x \in X$, diem que x *genera un cicle de longitud* e si x forma part d'un cicle de longitud e en la decomposició cíclica de F. És a dir, si la seqüència d'elements de X

$$x, F(x), F^2(x), \ldots, F^k(x), \ldots$$

té període e. En aquest cas diem que x és una *llavor* que genera el cicle.

Exemple 2.4. Si considerem el conjunt $A = \{1, 2, 3, 4, 5, 6\}$, una possible bijecció (o permutació) $F : A \to A$ seria la següent:

$$\begin{aligned} F(1) &= 4 \qquad & F(4) &= 2 \\ F(2) &= 1 \qquad & F(5) &= 5 \\ F(3) &= 6 \qquad & F(6) &= 3 \end{aligned}$$

A la Figura 2.2 podem veure gràficament la descomposició de F en cicles. L'estructura cíclica de F és doncs $\mathcal{CS}(F) = \{1, 2, 3\}$. Observem que l'element 4 del conjunt és una llavor que genera un cicle de longitud 3.

Ara considerem un digraf $G = (V, E)$. Direm que una permutació del conjunts de vèrtexs de G, $F : V \to V$, és *compatible amb l'estructura de* G si per a tot $x \in V$ es té que $(x, F(x)) \in E$. Si F és una permutació del conjunt de vèrtexs V compatible amb l'estructura de G, aleshores la descomposició

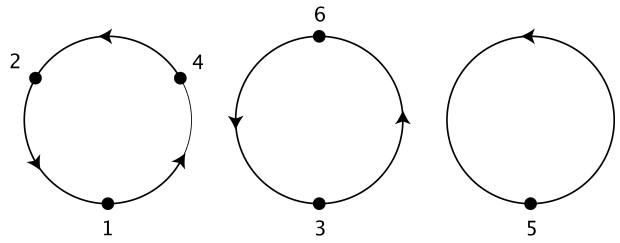

Figura 2.2: Descomposició cíclica de la permutació F.

de F en cicles és compatible amb l'estructura del graf G. Dit d'una altra manera, F indueix una descomposició del graf G en cicles vèrtex-disjunts.

D'aquesta manera obtenim una nova reducció. Podem resoldre el problema de trobar un cicle sobre G de longitud e si trobem una bijecció de V compatible amb l'estructura de G de manera que en la seva descomposició en cicles aparegui un cicle de longitud e. És a dir, busquem una bijecció $F : V \to V$ tal que per a tot $x \in V$ es té $(x, F(x)) \in E$ i de manera que tinguem $e \in \mathcal{CS}(F)$.

Vist d'aquesta manera, podem plantejar el següent problema, la solució del qual ens permetria resoldre també el Problema 2.

Problema 3. Donats dos enters positius m i e i agafant $n = \lceil \log_m e \rceil$, trobar de manera eficient una permutació F del conjunt de vèrtexs del digraf $B(m, n)$ compatible amb l'estructura del digraf de manera que tinguem $e \in \mathcal{CS}(F)$. A més, trobar una llavor $v \in V(B(m, n))$ que generi un cicle de F de longitud e.

A primera vista, el problema de donar una bijecció que contingui el cicle que estem buscant pot semblar més difícil que el problema original. No obstant, en les seccions que venen veurem que si ens restringim a un tipus de m concrets podem usar permutacions lineals dels vèrtexs per resoldre el problema.

2.5 Àlgebra lineal sobre cossos finits

En general, si el conjunt V de vèrtexs d'un graf G té cardinal $|V|$ tenim $|V|!$ permutacions diferents de V. D'aquestes, algunes respectaran l'estructura

de G i algunes no, i, a priori, és difícil classificar-les. A nosaltres ens interessa treballar només amb aquelles que sí respectin l'estructura de G. Per a una subfamília concreta dels grafs de deBruijn, l'àlgebra lineal dóna eines molt útils per estudiar un conjunt de permutacions que són compatibles amb l'estructura d'aquests digrafs.

Considerem un graf de deBruijn de la forma $B(q,n)$ on $q = p^k$ amb p un nombre primer i k un enter positiu. En aquest cas l'alfabet M sobre el que està definit el graf té q símbols i es pot dotar d'estructura de cos finit. Així ho farem i de fet identificarem l'alfabet amb $\mathbb{F}_q$, el cos finit de q elements. Per aquests grafs el conjunts de vèrtexs $V = M^n = \mathbb{F}_q^n$ es pot dotar d'estructura d'espai vectorial. Aleshores podem considerar la subfamília de bijeccions de V que preserven l'estructura d'espai vectorial: les aplicacions lineals bijectives de $\mathbb{F}_q^n$. Si a més imposem que aquestes aplicacions respectin l'estructura de $B(q,n)$, aleshores obtenim aplicacions d'una forma molt particular.

Recordem que tota aplicació lineal d'un espai vectorial de dimensió n en ell mateix es pot representar en una base donada amb una matriu $n \times n$ amb coeficients sobre el cos base de l'espai vectorial. En el nostre cas, una matriu $n \times n$ amb coeficients a $\mathbb{F}_q$. Sigui $F : \mathbb{F}_q^n \to \mathbb{F}_q^n$ una aplicació lineal bijectiva. Si F respecta l'estructura de $B(q,n)$ aleshores significa que per a tot $x \in \mathbb{F}_q^n$ tenim $x \to F(x)$. Per definició de la relació de desplaçament això vol dir que

$$(x_1, \ldots, x_{n-1}) = (y_0, \ldots, y_{n-2})$$

on $x = (x_0, x_1, \ldots, x_{n-1})$ i $F(x) = (y_0, y_1, \ldots, y_{n-1})$. En termes de la matriu de F expressada en la base canònica això vol dir que serà de la forma

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & a_0 \\ 1 & 0 & \cdots & 0 & a_1 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & a_{n-2} \\ 0 & 0 & \cdots & 1 & a_{n-1} \end{pmatrix},$$

amb $a_0, a_1, \ldots, a_{n-1} \in \mathbb{F}_q$. Prenem la convenció de considerar els vectors de $\mathbb{F}_q^n$ com a files que es multipliquen a la matriu per l'esquerra. A partir d'ara identificarem una aplicació F amb la seva matriu en la base canònica i les representarem amb el mateix símbol.

Observem que desenvolupant per la primera fila podem calcular fàcilment el determinant de F i obtenim

$$\det(F) = a_0.$$

Per tant l'aplicació serà bijectiva si i només si $a_0 \neq 0$. El càlcul del polinomi característic de F dóna

$$\mathrm{car}(F) = x^n - (a_{n-1}x^{n-1} + \cdots + a_1 x + a_0).$$

Per tant la matriu F queda unívocament determinada pel seu polinomi característic. De fet, donat un polinomi mònic $a(x)$ de grau n sempre es pot construir una matriu d'aquesta forma que té $a(x)$ per polinomi característic. La matriu s'anomena l'*acompanyant* de $a(x)$. La matriu acompanyant de $a(x)$ és invertible si i només si $a(0) \neq 0$.

Exemple 2.5. Considerem el digraf $B(3,2)$ sobre l'alfabet $\mathbb{F}_3 = \{0,1,2\}$. Calcularem l'estructura cíclica de la matriu acompanyat del polinomi $a(x) = x^2 - 2 \in \mathbb{F}_3[X]$,

$$F = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}.$$

Cadascun dels vectors de $\mathbb{F}_3^2$ estarà en un i només un dels cicles en què descomposa F. Prenent $v_0 = (0,0)$ tenim $v_0F = v_0$ i per tant v_0 està en un cicle de longitud 1. Prenent $v_1 = (1,0)$ podem calcular la seqüència v_1F, v_1F^2, ..., fins que tornem a v_1. Això ens dóna

$$(0,2), (2,0), (0,1), (1,0)$$

i per tant v_1, v_1F, v_1F^2 i v_1F^3 formen un cicle de longitud 4. Ara prenem $v_2 = (1,1)$ i repetint el mateix càlcul que en el cas anterior obtenim

$$(1,2), (2,2), (2,1), (1,1).$$

Així doncs v_2 dóna lloc a un altre cicle de longitud 4. Com que a $\mathbb{F}_q^2$ només hi ha 9 vectors, ja els hem vist tots i per tant tenim $\mathcal{CS}(F) = \{1,4\}$. Observem que podíem haver pres per v_1 qualsevol dels vectors del cicle que genera v_1 i el mateix per a v_2.

Més endavant veurem que el polinomi característic juga un paper molt

important en l'estudi de l'estructura cíclica de F. Això és degut a les propietats que té l'aplicació transposada de F que estudiem a continuació.

Tota aplicació lineal $F : V \to W$ té associada una aplicació $F^T : W \to V$ anomenada l'*aplicació transposada* de F. La matriu de F^T és, tal com el seu nom indica, la transposada de la matriu de F. En el cas que ens ocupa, si $F : \mathbb{F}_q^n \to \mathbb{F}_q^n$ és de la forma considerada, llavors F^T té matriu

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \end{pmatrix}.$$

El següent és un conegut resultat d'àlgebra lineal que ens serà útil.

Proposició 2.5. *Si M és una matriu $n \times n$, llavors M i M^T són similars. És a dir, existeix una matriu P invertible de manera que $M^T = PMP^{-1}$.*

Per a les aplicacions F que estem considerant podem donar la matriu P de manera explícita. En general aquesta matriu no és única, però aquí en donarem una que és fàcil de calcular ja que més endavant ens serà útil.

Proposició 2.6. *Sigui M la matriu acompanyant d'un polinomi mònic $a(x)$ de grau n tal que $a(0) \neq 0$. Aleshores la matriu*

$$P = \begin{pmatrix} 0 & \cdots & 0 & p_0 \\ \vdots & \iddots & \iddots & p_1 \\ 0 & p_0 & \iddots & \vdots \\ p_0 & p_1 & \cdots & p_{n-1} \end{pmatrix}, \qquad \text{on} \qquad \begin{aligned} p_0 &= 1 \\ p_k &= \sum_{i=1}^{k} p_{k-i} a_{n-i} \end{aligned}$$

per a $1 \leq k \leq n-1$, és invertible i compleix $PMP^{-1} = M^T$.

Demostració. Primer de tot calculem

$$PM = \begin{pmatrix} 0 & \cdots & 0 & p_0 & p_0a_{n-1} \\ \vdots & \iddots & p_0 & p_1 & p_0a_{n-2} + p_1a_{n-1} \\ 0 & \iddots & p_1 & p_2 & p_0a_{n-3} + p_1a_{n-2} + p_2a_{n-1} \\ p_0 & \iddots & \iddots & \vdots & \vdots \\ p_1 & p_2 & \cdots & p_{n-1} & p_0a_0 + \cdots + p_{n-1}a_{n-1} \end{pmatrix}$$

$$= \begin{pmatrix} 0 & \cdots & 0 & p_0 & p_1 \\ \vdots & \iddots & p_0 & p_1 & p_2 \\ 0 & \iddots & p_1 & p_2 & p_3 \\ p_0 & \iddots & \iddots & \vdots & \vdots \\ p_1 & p_2 & \cdots & p_{n-1} & p_0a_0 + \cdots + p_{n-1}a_{n-1} \end{pmatrix},$$

on l'última igualtat és deguda a les condicions imposades sobre els p_k per a $1 \leq k \leq n-1$. Com que tant P com PM són simètriques tenim que $PM = (PM)^T = M^T P^T = M^T P$. Per ser $p_0 = 1$ la matriu P és invertible i hem acabat. □

La propietat fonamental de l'aplicació transposada per als notres propòsits és la següent.

Proposició 2.7. *Si $F : \mathbb{F}_q^n \to \mathbb{F}_q^n$ és una aplicació lineal bijectiva, llavors $\mathcal{CS}(F) = \mathcal{CS}(F^T)$.*

Demostració. Sigui $e \in \mathcal{CS}(F)$. Aleshores en la descomposició cíclica de F tenim un cicle de longitud e i per tant existeix $v \in \mathbb{F}_q^n$ tal que v, vF, vF^2, …, vF^{e-1} són tots diferents i $v = vF^e$. Considerem el vector $u = vP^{-1}$ on P és tal que $F^T = PFP^{-1}$. Veurem que els vectors u, uF^T, $u(F^T)^2$, …, $u(F^T)^{e-1}$ són tots diferents i que a més $u = u(F^T)^e$. Per reducció a l'absurd, suposem que tenim una repetició: $u(F^T)^i = u(F^T)^j$ per alguns $0 \leq i < j \leq e-1$. Aleshores, per ser F^T invertible, $u = u(F^T)^{j-i}$. Per tant u serà el primer a repetir-se. Però si $u = u(F^T)^k$ amb $k \leq e-1$, llavors $vP^{-1} = vP^{-1}(F^T)^k$ i per tant $v = vF^k$, que és una contradicció. Per tant els vectors u, uF^T, $u(F^T)^2$, …, $u(F^T)^{e-1}$ són tots diferents. A més, per ser $v = vF^e = vP^{-1}(F^T)^e P$, tenim $u = u(F^T)^e$ tal com volíem. Per la invertibilitat de totes les matrius

implicades, l'argument és completament reversible i tenim la igualtat de les dues estructures cícliques. □

En general, el mateix argument permet demostrar que dues matrius quadrades invertibles i similars tenen la mateixa estructura cíclica, és a dir, que l'estructura cíclica és invariant per canvi de base. Per tant l'estructura cíclica és un concepte ben definit associat a l'aplicació i és independent de la matriu que prenguem per representar l'aplicació. Per tant, com que el polinomi característic també és un invariant de la matriu per canvi de base, ens plantegem estudiar la relació entre l'estructura cíclica i el polinomi característic. Per fer-ho, veurem que l'acció de F^T sobre un vector de $\mathbb{F}_q^n$ es pot relacionar amb una operació sobre l'anell de polinomis $\mathbb{F}_q[X]/(a(x))$, on $a(x)$ és el polinomi característic de F.

Sigui $u = (u_0, \ldots, u_{n-1}) \in \mathbb{F}_q^n$ i anomenem $w = uF^T$. Un senzill càlcul ens dóna

$$w = (w_0, w_1, \ldots, w_{n-1}) = (0, u_0, \ldots, u_{n-2}) + u_{n-1}(a_0, a_1, \ldots, a_{n-1}).$$

Per altra banda, si considerem els polinomis de $\mathbb{F}_q[X]$,

$$\begin{aligned} u(x) &= u_0 + u_1 x + \cdots + u_{n-1}x^{n-1}, \text{ i} \\ w(x) &= w_0 + w_1 x + \cdots + w_{n-1}x^{n-1}, \end{aligned}$$

podem utilitzar la relació entre u i w per veure que es compleix la relació

$$w(x) = xu(x) - u_{n-1}a(x),$$

on $a(x) \in \mathbb{F}_q[X]$ és el polinomi característic de la matriu associada a F,

$$a(x) = x^n - (a_{n-1}x^{n-1} + \cdots a_1 x + a_0).$$

Una altra manera més convenient d'expressar aquesta relació és

$$w(x) = (xu(x) \mod a(x)).$$

Fent ús d'aquesta equació tenim una nova manera de mirar-nos l'aplicació F com a una aplicació entre anells de polinomis:

$$F : \mathbb{F}_q[X]/(a(x)) \to \mathbb{F}_q[X]/(a(x))$$

de manera que $F(u(x)) = xu(x)$. Com que aquesta aplicació queda completament determinada pel polinomi $a(x)$, a partir d'ara denotarem la seva estructura cíclica per $\mathcal{CS}(a(x))$. Cadascun dels elements d'aquesta estructura cíclica és la longitud d'una seqüència de polinomis de $\mathbb{F}_q[X]/(a(x))$ donada per una condició inicial $u(x) \in \mathbb{F}_q[X]/(a(x))$.

Exemple 2.6. Considerem el digraf $B(3,2)$ sobre l'alfabet $\mathbb{F}_3 = \{0,1,2\}$. A l'Exemple 2.5 hem vist que el vector $v = (1,0) \in \mathbb{F}_3^2$ dóna lloc a un cicle de longitud 4 sota l'acció de la matriu acompanyant de $a(x) = x^2 - 2 \in \mathbb{F}_3[X]$. Veurem que el mateix passa en termes de polinomis. Si F és la matriu acompanyant de $a(x)$ i F^T la seva tranposada, aleshores per la Proposició 2.6 tenim $F^T = PFP^{-1}$ on P és la matriu

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

El vector $u = vP^{-1} = (0,1)$ ens defineix el polinomi $u(x) = 1$ de $\mathbb{F}_3[X]/(a(x))$. Calculant els polinomis $xu(x)$, $x^2u(x)$, ... mòdul $a(x)$ fins que tornem a trobar $u(x)$ obtenim: x, 2, $2x$ i 1. Per tant la longitud del cicle que conté $u(x)$ en la descomposició cíclica de l'aplicació F^T és 4, igual que la del cicle que conté v en la descomposició de F.

La conclusió que treiem d'aquesta secció és que, si, enlloc de considerar el Problema 3 en general, considerem la versió restringida que enunciem a continuació en el Problema 4, aleshores és possible usar mètodes algebraics per estudiar el problema. Això és el que pensem fer en les pròximes seccions.

Problema 4. Donats dos enters positius q i e amb $q = p^k$ per algun primer p i agafant $n = \lceil \log_q e \rceil$, trobar de manera eficient un polinomi $a(x) \in \mathbb{F}_q[X]$ de grau n de manera que tinguem $e \in \mathcal{CS}(a(x))$. A més, trobar una llavor $u(x) \in \mathbb{F}_q[X]$ de grau menor que n que generi un cicle de longitud e de $a(x)$.

2.6 Polinomis sobre cossos finits

Hem vist que els polinomis amb coeficients en un cos finit jugaran un paper important en l'estudi de les estructures cícliques que ens interessen. Això serà a les pròximes seccions. Abans però, dedicarem aquesta secció a recollir un seguit de resultats algebraics que ens seran útils més endavant. Tots els

resultats que s'enuncien sense demostració es poden trobar a LIDL i NIEDERREITER (1997, Capítol 3).

Considerem $\mathbb{F}_q$ un cos finit arbitrari però fix de característica p. Denotarem per $\mathbb{F}_q[X]$ l'anell de polinomis en una variable amb coeficients en el cos $\mathbb{F}_q$. Sigui $a(x) \in \mathbb{F}_q[X]$ un polinomi de grau n, aleshores denotem per $\mathbb{F}_q[X]/(a(x))$ l'anell de polinomis mòdul $a(x)$. Tots els polinomis d'aquest anell tenen grau estrictament menor que n i, per ser $\mathbb{F}_q$ un cos finit, n'hi ha un nombre finit. Exactament q^n, dels quals $q^n - 1$ són no nuls.

Si prenem la seqüència formada pels polinomis

$$x^s \mod a(x), \text{ per } 0 \leq s \leq q^n - 1$$

tenim q^n elements no nuls de $\mathbb{F}_q[X]/(a(x))$. Com que no poden ser tots diferents, hi ha d'haver alguna repetició:

$$x^{s_1} \equiv x^{s_2} \mod a(x), \text{ on } 0 \leq s_1 < s_2 \leq q^n - 1.$$

Per tant $a(x)$ divideix $x^{s_2} - x^{s_1} = x^{s_1}(x^{s_2-s_1} - 1)$ i per ser irreductible aleshores dividirà algun dels dos factors. Si imposem que $a(0) \neq 0$, aleshores $a(x)$ no pot dividir x^{s_1} i per tant $a(x)$ ha de dividir $x^{s_2-s_1} - 1$. Això ens porta a la següent definició: si $a(x)$ és un polinomi tal que $a(0) \neq 0$, el menor enter e tal que $a(x)$ divideix $x^e - 1$ s'anomena l'*ordre* de $a(x)$. Dit d'una altra manera, e és el mínim enter tal que $x^e \equiv 1 \mod a(x)$. Denotarem l'ordre d'un polinomi $a(x)$ per $\text{ord}(a(x))$. Per definició tenim doncs $\text{ord}(a(x)) \in \mathcal{CS}(a(x))$.

Més avall recollim una sèrie de coneguts resultats sobre les propietats de l'ordre. Però abans es fa necessari introduir dos nous elements de notació.

Recordem que si dos enters $a, b \geq 2$ són coprimers entre ells, aleshores un és invertible mòdul l'altre. En aquest cas té sentit definir l'*ordre de a mòdul b* que denotem com $\text{ord}_b(a)$ i que és el menor enter positiu i tal que $a^i \equiv 1 \mod b$.

Si p és un nombre primer i s un enter positiu, denotarem la part entera superior del logaritme de s en base p mitjançant el símbol $\lceil s \rceil_p$. És a dir, $\lceil s \rceil_p = \lceil \log_p s \rceil$. Quan no hi hagi possibilitat de confusió sobre el primer en qüestió, cometrem un abús de notació i escriurem $\lceil s \rceil$. Observem que sempre tenim $p^{\lceil s \rceil - 1} < s \leq p^{\lceil s \rceil}$, o sigui, $\lceil s \rceil$ és el mínim enter tal que $s \leq p^{\lceil s \rceil}$. La Taula 2.3 recull uns quants valors de $\lceil s \rceil$ per $p = 2$.

El primer resultat que enunciem recull una propietat que relaciona l'ordre d'un polinomi amb el seu grau i la cardinalitat del cos on estan els coeficients.

s	$\lceil s \rceil$	s	$\lceil s \rceil$
1	0	6	3
2	1	7	3
3	2	8	3
4	2	9	4
5	3	10	4

Taula 2.3: Els deu primers valors de $\lceil s \rceil_2$

Proposició 2.8. *L'ordre d'un polinomi irreductible $a(x) \in \mathbb{F}_q[X]$ de grau n i tal que $a(0) \neq 0$ sempre és un divisor de $q^n - 1$. En particular, no és múltiple de p.*

Proposició 2.9. *El màxim comú divisor de $x^r - 1$ i $x^s - 1$ és $x^{\mathrm{mcd}(r,s)} - 1$. A més, si $a(x) \in \mathbb{F}_q[X]$ és tal que $a(0) \neq 0$, aleshores $a(x)$ divideix $x^s - 1$ si i només si $\mathrm{ord}(a(x))$ divideix s.*

Un cas particular d'aquest resultat que ens serà útil és que $x^r - 1$ divideix $x^s - 1$ si i només si r divideix s.

Proposició 2.10. *Sigui $e \geq 2$ un enter coprimer amb q. Aleshores existeixen en $\mathbb{F}_q[X]$ polinomis irreductibles d'ordre e. A més, tots tenen grau $\mathrm{ord}_e(q)$.*

Observem que si $a(x)$ és un polinomi irreductible amb $a(0) \neq 0$ tal que $\mathrm{ord}(a(x)) = e$, aleshores, per la Proposició 2.9, $a(x)$ divideix $x^e - 1$. I de fet, tots els polinomis irreductibles que són divisors de $x^e - 1$ tenen per ordre algun divisor de e. Això ens dóna un mètode per trobar els polinomis de la Proposició 2.10. Considerem el polinomi

$$a_e(x) = \frac{x^e - 1}{\mathrm{mcm}_{d|e, d \neq e} \{x^d - 1\}} \in \mathbb{F}_q[X].$$

Qualsevol factor irreductible de $a_e(x)$ tindrà ordre e ja que serà un divisor de $x^e - 1$ però no ho serà de $x^d - 1$ per a cap d divisor de e. Per tant, calcular $a_e(x)$ i factoritzar-lo és un mètode per a obtenir polinomis irreductibles d'un ordre e donat.

Proposició 2.11. *Sigui $a(x) \in \mathbb{F}_q[X]$ un polinomi irreductible d'ordre e tal que $a(0) \neq 0$. Aleshores es compleix $\mathrm{ord}(a(x)^s) = ep^{\lceil s \rceil}$.*

Proposició 2.12. *Siguin* $a_1(x), \ldots, a_r(x) \in \mathbb{F}_q[X]$ *polinomis dos a dos coprimers tals que* $a_i(0) \neq 0$*, i siguin* $e_i = \operatorname{ord}(a_i(x))$ *els seus respectius ordres. Aleshores* $\operatorname{ord}(a_1(x) \cdots a_r(x)) = \operatorname{mcm}\{e_1, \ldots, e_r\}$.

Aquests dos últims resultats donen un mètode per calcular l'ordre d'un polinomi compost en funció dels ordres dels polinomis irreductibles en què descomposa.

Exemple 2.7. Considerem el següent polinomi $a(x) \in \mathbb{F}_2[X]$ i la seva factorització en polinomis irreductibles de $\mathbb{F}_2[X]$:

$$a(x) = x^8 + x^7 + x^6 + x^5 + x^4 + 1 = (x^3 + x^2 + 1)(x^2 + x + 1)(x + 1)^3.$$

Per la Proposició 2.12 tenim que $\operatorname{ord}(a(x)) = \operatorname{mcm}\{e_1, e_2, e_3\}$ on

$$\begin{aligned} e_1 &= \operatorname{ord}(x^3 + x^2 + 1) \\ e_2 &= \operatorname{ord}(x^2 + x + 1) \\ e_3 &= \operatorname{ord}((x + 1)^3) \end{aligned}$$

Calculant obtenim $e_1 = 7$ i $e_2 = 3$. Per altra banda, fent ús de la Proposició 2.11, com que $\operatorname{ord}(x + 1) = 1$ i $\lceil 3 \rceil_2 = 2$ aleshores tenim $e_3 = 1 \cdot 2^2 = 4$. Per tant

$$\operatorname{ord}(a(x)) = \operatorname{mcm}\{7, 3, 4\} = 84.$$

A part de la seva aplicació al càlcul d'ordres de polinomis, els resultats anteriors permeten provar un important resultat teòric.

Corol·lari 2.13. *Per a tot enter* $e \geq 1$ *existeix un polinomi* $a(x) \in \mathbb{F}_q[X]$ *amb* $\operatorname{ord}(a(x)) = e$.

Demostració. Per a $e = 1$ es té $\operatorname{ord}(x - 1) = 1$. Sigui ara $e \geq 2$. Si e i q són coprimers, aleshores per la Proposició 2.10 existeix $a(x) \in \mathbb{F}_q[X]$ irreductible tal que $\operatorname{ord}(a(x)) = e$. Per tant suposem que e i q no són coprimers. Aleshores podem escriure e com $e = p^\alpha e_*$ de manera que $\alpha \geq 1$ i e_* i q són coprimers. Ara, per la Proposició 2.11, tenim $\operatorname{ord}(b(x)) = p^\alpha$ on

$$b(x) = (x - 1)^{p^{\alpha-1}+1}$$

ja que $\lceil p^{\alpha-1} + 1 \rceil = \alpha$. Si prenem $a_*(x) \in \mathbb{F}_q[X]$ un polinomi irreductible d'ordre e_*, agafant $a(x) = b(x)a_*(x)$ tindrem, per la Proposició 2.12,

$$\operatorname{ord}(a(x)) = \operatorname{mcm}\{p^\alpha, e_*\} = p^\alpha e_* = e. \qquad \square$$

Per acabar enunciarem i demostrarem un lema tècnic que ens permetrà establir relacions entre els graus de dos polinomis a partir de relacions entre els seus ordres.

Lema 2.14. *Siguin $a, b, q \geq 2$ tres enters de manera que a i b són coprimers amb q. Aleshores,*

$$\mathrm{ord}_{\mathrm{mcm}\{a,b\}}(q) = \mathrm{mcm}\left\{\mathrm{ord}_a(q), \mathrm{ord}_b(q)\right\}.$$

En particular,

1. *si a divideix b llavors $\mathrm{ord}_a(q)$ divideix $\mathrm{ord}_b(q)$,*

2. *si a i b són coprimers llavors $\mathrm{ord}_{ab}(q) = \mathrm{mcm}\left\{\mathrm{ord}_a(q), \mathrm{ord}_b(q)\right\}$.*

Demostració. Denotarem per e_a, e_b i $e_{a,b}$ els ordres de q mòdul a, b i $\mathrm{mcm}\{a, b\}$ respectivament. D'una banda tenim que a divideix $\mathrm{mcm}\{a, b\}$ que divideix $q^{e_{a,b}} - 1$ i per tant $q^{e_{a,b}} \equiv 1 \mod a$. Per definició d'ordre tindrem doncs que e_a divideix $e_{a,b}$. El mateix argument prova que e_b també divideix $e_{a,b}$ i per tant $\mathrm{mcm}\{e_a, e_b\}$ divideix $e_{a,b}$. D'altra banda, de $q^{e_a} \equiv 1 \mod a$ i $q^{e_b} \equiv 1 \mod b$ obtenim, respectivament, $q^{\mathrm{mcm}\{e_a,e_b\}} \equiv 1 \mod a$ i $q^{\mathrm{mcm}\{e_a,e_b\}} \equiv 1 \mod b$. Això implica que $q^{\mathrm{mcm}\{e_a,e_b\}} \equiv 1 \mod \mathrm{mcm}\{a, b\}$ i per definició d'ordre tenim que $e_{a,b}$ divideix $\mathrm{mcm}\{e_a, e_b\}$. Per tant $e_{a,b} = \mathrm{mcm}\{e_a, e_b\}$ tal com volíem. □

2.7 L'estructura cíclica

Fins ara hem vist que si tenim un digraf de deBruijn de la forma $B(q, n)$ on q és una potència d'un nombre primer, aleshores l'estructura cíclica d'una permutació dels vèrtexs de $B(q, n)$ donada per la matriu acompanyant d'un polinomi mònic $a(x) \in \mathbb{F}_q[X]$ de grau n i tal que $a(0) \neq 0$ coincideix amb l'estructura cíclica de la bijecció de $\mathbb{F}_q[X]/(a(x))$ donada per

$$\begin{array}{ccc} \mathbb{F}_q[X]/(a(x)) & \longrightarrow & \mathbb{F}_q[X]/(a(x)) \\ u(x) & \longmapsto & xu(x) \end{array}$$

L'estudi d'aquesta estructura cíclica, que denotem per $\mathcal{CS}(a(x))$ és l'objectiu d'aquesta secció.

La remarca que ens serveix de punt de partida per aquest estudi és la següent: donat un polinomi $a(x)$ amb coeficients sobre un cos $\mathbb{F}$ sempre el podem descomposar en un producte de la forma

$$a(x) = \alpha a_1(x)^{s_1} a_2(x)^{s_2} \cdots a_r(x)^{s_r}$$

on α és un element no nul del cos $\mathbb{F}$, s_i són enters positius i $a_i(x)$ són polinomis mònics irreductibles sobre el cos $\mathbb{F}$. Aquest fet fonamental ens està dient que les peces "bàsiques" de qualsevol polinomi són els polinomis mònics irreductibles. Com hem vist anteriorment, per les Proposicions 2.11 i 2.12, coneixent els ordres dels polinomis irreductibles $a_i(x)$ podíem calcular l'ordre del polinomi $a(x)$. L'objectiu que perseguim és generalitzar aquests resultats de manera que ens permetin calcular l'estructura cíclica del polinomis $a(x)$ a partir de les estructures cícliques dels seus factors irreductibles.

El primer pas és coneixer l'estructura cíclica de les peces "bàsiques", els polinomis mònics irreductibles.

Proposició 2.15. *Si $a(x)$ és un polinomi mònic irreductible d'ordre e, llavors $\mathcal{CS}(a(x)) = \{1, e\}$.*

Demostració. Prenent com a llavor $u(x) = 0$ està clar que tenim un cicle de longitud 1 i per tant $1 \in \mathcal{CS}(a(x))$. Com que $\mathrm{ord}(a(x)) = e$, la llavor $u(x) = 1$ genera un cicle de longitud e i per tant $e \in \mathcal{CS}(a(x))$. Sigui ara $u(x)$ un polinomi no nul de grau menor que el de $a(x)$ i sigui k la longitud del cicle que genera. O sigui, k és el mínim enter positiu tal que $u(x)x^k \equiv u(x)$ mòdul $a(x)$. Això ens diu que k és el mínim enter positiu tal que $a(x)$ divideix $u(x)(x^k - 1)$. Com que $a(x)$ és irreductible aleshores ha de dividir un dels dos factos. Com que $u(x)$ té grau menor que $a(x)$, $a(x)$ no pot dividir $u(x)$ i per tant ha de dividir $x^k - 1$. Així doncs, k és el mínim enter positiu tal que $a(x)$ divideix $x^k - 1$, però això és l'ordre de $a(x)$. Per tant, $k = e$ i no és possible cap altre longitud de cicle. □

El següent pas és veure com es comporta l'estructura cíclica d'un polinomi mònic irreductible quan prenem una potència d'aquest polinomi.

Proposició 2.16. *Sigui $a(x)$ un polinomi mònic irreductible d'ordre e. L'estructura cíclica de $a(x)^s$ és $\mathcal{CS}(a(x)^s) = \left\{1, e, ep, \ldots, ep^{\lceil s \rceil}\right\}$.*

Demostració. Prenent llavor $u(x) = 0$ està clar que $1 \in \mathcal{CS}(a(x)^s)$. Sigui ara $u(x) \in \mathbb{F}_q[X]$ no nul de grau menor que $a(x)^s$ i denotem per k la longitud

del cicle que genera. Això vol dir que k és el mínim enter positiu tal que $x^k u(x) \equiv u(x) \mod a(x)^s$, o d'una altra manera, el mínim enter positiu tal que $a(x)^s$ divideix $(x^k-1)u(x)$. Sigui $0 \leq d < s$ el màxim enter tal que $a(x)^d$ divideix $u(x)$, llavors podem posar $u(x) = a(x)^d u'(x)$ amb $u'(x) \in \mathbb{F}_q[X]$ coprimer amb $a(x)$. Vist així tenim que k és el mínim enter positiu tal que $a(x)^{s-d}$ divideix $x^k - 1$, i per definició això és l'ordre de $a(x)^{s-d}$. Per la Proposició 2.11 tenim $k = ep^{\lceil s-d \rceil}$ i per ser $s-d \leq s$ tenim que $\lceil s-d \rceil \leq \lceil s \rceil$. Concloem doncs que totes les longituds que apareixen a $\mathcal{CS}(a(x)^s)$ són de la forma considerada a l'enunciat. Per veure que totes les longituds considerades apareixen en $\mathcal{CS}(a(x)^s)$ és suficient observar que prenent $u(x) = a(x)^{s-1}$ tenim longitud e i que prenent $u(x) = a(x)^{s-(p^{i-1}+1)}$ per $1 \leq i \leq \lceil s \rceil$ tenim longitud ep^i. □

Usant aquesta proposició estem en condicions de provar el següent resultat que caracteritza completament l'estructura cíclica d'un polinomi mònic tal que $a(0) \neq 0$. Per fixar la notació, suposarem que la factorització de $a(x)$ vé donada per

$$a(x) = a_1(x)^{s_1} a_2(x)^{s_2} \cdots a_r(x)^{s_r}$$

de manera que es compleixi $s_1 \leq s_2 \leq \cdots \leq s_r$. El conjunt d'índexs el denotarem per $I = \{1, \ldots, r\}$.

Proposició 2.17. *Sigui $a(x)$ un polinomi mònic amb $a(0) \neq 0$ factoritzat tal com hem indicat. Si denotem $e_i = \operatorname{ord}(a_i(x))$, aleshores l'estructura cíclica de $a(x)$ vé donada per*

$$\mathcal{CS}(a(x)) = \{1\} \bigcup \left\{ p^t \operatorname{mcm}_{i \in J} \{e_i\} \mid \emptyset \neq J \subseteq I,\ 0 \leq t \leq \lceil s_j \rceil,\ j = \max J \right\}.$$

Demostració. Prenent $u(x) = 0$ veiem que $1 \in \mathcal{CS}(a(x))$. Prenem ara $u(x) \in \mathbb{F}_q[X]$ un polinomi no nul de grau menor que $a(x)$ i que generi un cicle de longitud $k \geq 2$ mòdul $a(x)$. Denotem per k_i la longitud de cicle que genenera $u(x)$ mòdul $a_i(x)^{s_i}$, és a dir, k_i és el mínim enter positiu tal que $x^{k_i} u(x) \equiv u(x) \mod a_i(x)^{s_i}$.

Vegem que $k = \operatorname{mcm}_{i \in I}\{k_i\}$. Per una banda tenim que, per a qualsevol $i \in I$, $a_i(x)^{s_i}$ divideix $u(x)(x^{\operatorname{mcm}_{i \in I}\{k_i\}} - 1)$. Per tant $a(x) = \operatorname{mcm}_{i \in I}\{a_i(x)^{s_i}\}$ divideix $u(x)(x^{\operatorname{mcm}_{i \in I}\{k_i\}} - 1)$ i com que k és l'ordre de $u(x)$ mòdul $a(x)$ tenim que k ha de dividir $\operatorname{mcm}_{i \in I}\{k_i\}$. Per altra banda, $a_i(x)^{s_i}$ divideix $a(x)$ que al seu torn divideix $u(x)(x^k - 1)$. Però k_i és el mínim enter positiu tal que

$a_i(x)^{s_i}$ divideix $u(x)(x^{k_i}-1)$, per tant k_i ha de dividir k i obtenim el que volíem.

Ara tenim que, per la Proposició 2.16, o bé $k_i = 1$ o bé $k_i = e_i p^j$ per algun $0 \leq j \leq \lceil s_i \rceil$. Així doncs, si denotem $J = \{i \in I \mid k_i \neq 1\}$, tenim $J \neq \emptyset$ ja que hem assumit $k \geq 2$ i per tant $k = \operatorname{mcm}_{i\in J}\{k_i\} = p^t \operatorname{mcm}_{i\in J}\{e_i\}$ per algun t tal que $0 \leq t \leq \lceil s_j \rceil$ on $j = \max J$ ja que hem pres els s_i ordenats de manera creixent.

Finalment manca veure que totes les longituds considerades apareixen com a longitud del cicle generat per alguna llavor. Donats un conjunt $J \subseteq I$ no buit i un enter t tal que $0 \leq t \leq \lceil s_j \rceil$ on $j = \max J$, definim el polinomi

$$u_{J,t}(x) = a_j(x)^{s_j-(p^{t-1}+1)} \left(\prod_{i\in J\setminus\{j\}} a_i(x)^{s_i-1} \right) \left(\prod_{i\notin J} a_i(x)^{s_i} \right),$$

on assumim $p^{-1} = 0$. Sigui k el mínim enter pel qual $a(x)$ divideix $u_{J,t}(x)(x^k - 1)$. Aquest és també el mínim k pel qual $a'(x)$ divideix $x^k - 1$ on $a'(x)$ és

$$a'(x) = \frac{a(x)}{u_{J,t}(x)} = \left(\prod_{i\in J\setminus\{j\}} a_i(x) \right) a_j(x)^{p^{t-1}+1}.$$

Per tant k és l'ordre de $a'(x)$, i aplicant la Proposició 2.12 tenim

$$k = \operatorname{ord}(a'(x)) = p^t \operatorname{mcm}_{i\in J}\{e_i\}$$

tal com volíem ja que $\lceil p^{t-1} + 1 \rceil = t$. □

Usant aquest resultat podem, per exemple, calcular l'estructura cíclica del polinomi de l'Exemple 2.7 del qual ja haviem calculat l'ordre.

Exemple 2.8. Considerem el polinomi $a(x) \in \mathbb{F}_2[X]$ de l'Exemple 2.7 donat per

$$a(x) = a_1(x)a_2(x)a_3(x)^3 = (x^3 + x^2 + 1)(x^2 + x + 1)(x + 1)^3.$$

Teniem $e_1 = 7$, $e_2 = 3$ i $e_3 = 1$. Usant la Proposició 2.17 obtenim

$$\mathcal{CS}(a(x)) = \{1, 2, 3, 6, 7, 12, 14, 21, 28, 42, 84\}.$$

Un cas particular important de la Proposició 2.17 que l'exemple posa de manifest i que ens serà útil més endavant és el següent.

Corol·lari 2.18. *Per a tot polinomi mònic* $a(x) \in \mathbb{F}_q[X]$ *tal que* $a(0) \neq 0$ *es té* $\operatorname{ord}(a(x)) \in \mathcal{CS}(a(x))$. *A més,* $\operatorname{ord}(a(x)) = \max \mathcal{CS}(a(x))$.

Demostració. La pertinença és clara, ja que prenent llavor $u(x) = 1$ obtenim $\operatorname{ord}(a(x))$ per definició. Per altra banda, com que els factors de $a(x)$, $a_1(x)^{s_1}, \ldots, a_r(x)^{s_r}$ són coprimers dos a dos, tenim que

$$\begin{aligned}\operatorname{ord}(a(x)) &= \operatorname{ord}\left(a_1(x)^{s_1} \cdots a_r(x)^{s_r}\right) \\ &= \operatorname{mcm}\left\{\operatorname{ord}\left(a_1(x)^{s_1}\right), \ldots, \operatorname{ord}\left(a_r(x)^{s_r}\right)\right\} \\ &= \operatorname{mcm}\left\{e_1 p^{\lceil s_1 \rceil}, \ldots, e_r p^{\lceil s_r \rceil}\right\} \\ &= p^{\lceil s_r \rceil} \operatorname{mcm}\left\{e_1, \ldots, e_r\right\},\end{aligned}$$

on l'última igualtat és deguda al fet que els ordres e_i dels polinomis són coprimers amb p i que $s_1 \leq \ldots \leq s_r$ implica que $\lceil s_1 \rceil \leq \ldots \leq \lceil s_r \rceil$ per la monotonia del logaritme i la part entera. Ara bé, el màxim de $\mathcal{CS}(a(x))$ es correspon amb la longtitud de cicle que s'obté quan s'agafa el conjunt d'índex $J = I$ i l'exponent $t = \lceil s_r \rceil$. Això dóna

$$\max \mathcal{CS}(a(x)) = p^{\lceil s_r \rceil} \operatorname{mcm}\left\{e_1, \ldots, e_r\right\}.$$

Per tant, tenim $\operatorname{ord}(a(x)) = \max \mathcal{CS}(a(x))$ tal com volíem. □

Per acabar aquesta secció, formalitzarem un altre resultat que es troba implícit en la demostració de la Proposició 2.17 i que juntament amb el Corol·lari anterior ens servirà de base per a la teoria que desenvoluparem en la pròxima secció.

Corol·lari 2.19. *Sigui* $a(x) \in \mathbb{F}_q[X]$ *un polinomi mònic tal que* $a(0) \neq 0$. *Si* e *és un enter diferent de* 1 *tal que* $e \in \mathcal{CS}(a(x))$, *aleshores tenim* $e = \operatorname{ord}(a'(x))$ *on* $a'(x)$ *és un divisor de* $a(x)$.

2.8 Polinomis d'ordre fixat i grau mínim

Hem acabat la secció anterior amb els Corol·laris 2.18 i 2.19. Aquests corol·laris ens diuen que l'ordre d'un polinomi és la màxima longitud que apareix en la seva estructura cíclica, i que si tenim una longitud que apareix en

l'estructura cíclica d'un cert polinomi $a(x)$, aleshores aquesta longitud és l'ordre d'algun divisor de $a(x)$. En altres paraules, ara sabem que l'únic terme rellevant en l'estructura cíclica d'un cert polinomi $a(x)$ és el seu ordre, ja que si tenim un polinomi que té en la seva estructura cíclica la longitd que estem bucant però aquesta longitud no és la màxima de l'estructura cíclica, aleshores podem trobar un polinomi de grau menor que té per ordre la longitud desitjada. La conclusió que treiem és que per resoldre el Problema 4 és suficient resoldre el següent problema més restringit.

Problema 5. Donats dos enters positius q i e amb $q = p^k$ per algun primer p i agafant $n = \lceil \log_q e \rceil$, trobar de manera eficient un polinomi $a(x) \in \mathbb{F}_q[X]$ de grau n i ordre e.

Observem que en aquest cas, si $a(x)$ és una solució al Problema 5, alehores tenim $\text{ord}(a(x)) = e$ i per tant podem prendre com a llavor $u(x) = 1$ per generar un cicle de longitud e. Per tant no cal buscar la llavor.

En tots els problemes que hem plantejat fins ara, hem demanat que donats q i e la solució ens permetés contruir una (q, n, e)-seqüència tancada amb $n = \lceil \log_q e \rceil$. L'existència d'aquesta seqüència la tenim garantida pel Teorema 2.2. No obstant, en general no podem afirmar que es pugui obtenir una d'aquestestes seqüències mitjançant permutacions lineals dels vèrtexs de $B(q, n)$. A continuació comentarem el perquè d'aquest fet.

Si fixem q i n, les longituds e tals que $n = \lceil \log_q e \rceil$ són totes les que satisfan

$$q^{n-1} + 1 \leq e \leq q^n.$$

En total, hi ha $q^{n-1}(q-1)$ longituds de seqüència diferents per a les quals caldria buscar una permutació dels vèrtexs de $B(q, n)$. De totes les permutacions possibles, nosaltres hem restringit el nostre domini de cerca a les permutacions lineals, i hem vist que cadascuna d'aquestes es correspon amb un polinomi mònic $a(x) \in \mathbb{F}_q[X]$ de grau n i tal que $a(0) \neq 0$. El total de polinomis d'aquest tipus també és $q^{n-1}(q-1)$. Vist això, un podria pensar que si cadascun d'aquests polinomis tingués un ordre diferent dins l'interval $[q^{n-1} + 1, q^n]$, aleshores per cadascuna de les longituds d'aquest interval podríem trobar un polinomi de grau n amb ordre igual a aquesta longitud. No obstant, en general, aquest no és el cas.

Un exemple el podem veure a la Taula 2.4 que recull els ordres de tots els polinomis $a(x) \in \mathbb{F}_2[X]$ mònics, de grau 4 i tals que $a(0) \neq 0$. En aquesta taula es pot apreciar com amb els 8 polinomis de grau 4 que hi ha només

$a(x)$	$\mathrm{ord}(a(x))$	Irreductible
x^4+1	4	No
x^4+x^3+1	15	Si
x^4+x^2+1	6	No
$x^4+x^3+x^2+1$	7	No
x^4+x+1	15	Si
x^4+x^3+x+1	6	No
x^4+x^2+x+1	7	No
$x^4+x^3+x^2+x+1$	5	Si

Taula 2.4: Ordres dels polinomis mònics de grau 4 de $\mathbb{F}_2[X]$

podem obtenir 5 longituds de cicle diferents i només una d'elles es troba en l'interval $[9, 16]$. Per tant, si volguessim, per exemple, construir un cicle de longitud $e = 10$ mitjançant una permutació lineal, ja sabem que no podríem assolir-ho amb un polinomi de grau $\lceil \log_2 10 \rceil = 4$. Caldria buscar entre els polinomis de grau més gran que 4.

Vist que en general el Problema 5 no té solució, ens decidirem per buscar una solució aproximada mitjançant els mètodes de què disposem. Formalment, ens proposem resoldre el següent problema.

Problema 6. Donats dos enters positius q i e amb $q = p^k$ per algun primer p, trobar de manera eficient un polinomi $a(x) \in \mathbb{F}_q[X]$ d'ordre e i grau mínim.

Observem que pel Corol·lari 2.13 sempre existirà algun polinomi d'ordre e i per tant té sentit buscar el de grau mínim. La demostració del corol·lari ens dóna una possible manera de construir un polinomi d'un ordre donat, però en general aquest no serà de grau mínim com el següent exemple posa de manifest.

Exemple 2.9. Considerem el cos $\mathbb{F}_2$ i la longitud $e = 210 = 2 \cdot 105$. Segons la Proposició 2.8 tenim en $\mathbb{F}_2[X]$ un polinomi irreductible d'ordre 105 i grau $\mathrm{ord}_{105}(2) = 12$. Usant el mètode que hem donat per calcular aquests polinomis obtenim el següent polinomi d'ordre 105:

$$a_*(x) = x^{12} + x^9 + x^5 + x^4 + x^3 + x + 1.$$

Per tant, el polinomi d'ordre 210 de la demostració del Corol·lari 2.13 seria

$$\begin{aligned} a(x) &= (x^{12} + x^9 + x^5 + x^4 + x^3 + x + 1)(x+1)^2 \\ &= x^{14} + x^{12} + x^{11} + x^9 + x^7 + x^6 + x^4 + x^2 + x + 1. \end{aligned}$$

Ara considerem el polinomi de $\mathbb{F}_2[X]$

$$b(x) = (x^3 + x^2 + 1)(x^4 + x^3 + 1)(x+1)^2 = x^9 + x^6 + x^5 + 1.$$

Usant que $\mathrm{ord}(x^3 + x^2 + 1) = 7$ i $\mathrm{ord}(x^4 + x^3 + 1) = 15$ podem calcular l'ordre de $b(x)$ i obtenim

$$\mathrm{ord}(b(x)) = \mathrm{mcm}\{7, 15, 2\} = 7 \cdot 15 \cdot 2 = 210.$$

Per tant $b(x)$ és un polinomi de grau menor que $a(x)$ i que també té ordre 210.

En general, donat un ordre e coprimer amb q, tampoc podem esperar que el grau mínim s'assoleixi en el polinomi irreductible d'aquest ordre que té grau $\mathrm{ord}_e(q)$.

Malgrat que la solució no sigui tant directa, la teoria desenvolupada en aquest capítol dóna eines suficients per resoldre el Problema 6 mitjançant un algorisme eficient. La formulació d'aquest algorisme, la demostració de la seva correctesa i l'anàlisi de la seva complexitat són els objectius del pròxim capítol.

Capítol 3

Construcció de q-seqüències de longitud arbitrària

Usant la teoria desenvolupada en el capítol anterior, en aquest capítol construirem un algorisme per resoldre el Problema 6.

En primer lloc estudiarem amb detall un exemple concret, el cas $q = 2$ i $e = 360$. Aquest cas és particularment útil ja que permet la construcció de codificadors angulars uni-corona amb resolució de 1º. Inspirats per aquest exemple, abordarem tot seguit el cas general amb la intenció de caracteritzar les possibles solucions del problema de manera que sigui fàcil considerar-les de manera algorísmica.

Amb els resultats que caracteritzen les possibles solucions a mà, passarem a la construcció de l'algorisme que dividirem en dues parts. Seguidament n'analitzarem la complexitat i justificarem que és més eficient que la cerca per força bruta que hem comentat a 2.2. Per acabar presentarem un resultat experimental que s'ha obtingut usant l'algorisme presentat.

3.1 Un exemple il·lustratiu

Tornant al problema de la codificació angular del Capítol 1, suposem que volem construir un codificador angular absolut amb resolució de 1º. Per fer-ho disposem de detectors binaris que volem disposar sobre una sola corona. Ens cal doncs construïr una $(2, n, 360)$-seqüència tancada amb la mínima n possible.

Com que 2 és primer, podem aplicar les tècniques desenvolupades en el

Capítol 2. En particular, volem resoldre el Problema 6 amb $q = 2$ i $e = 360$. És a dir, buscar un polinomi $a(x) \in \mathbb{F}_2[X]$ d'ordre 360 i grau mínim.

Com que 360 és divisible per 2, sabem per la Proposició 2.8 que cap polinomi irreductible de $\mathbb{F}_2[X]$ pot tenir ordre 360. Per tant caldrà buscar un polinomi que no sigui irreductible. Suposem que $a(x) \in \mathbb{F}_2[X]$ és un polinomi compost d'ordre 360, grau n i tal que $a(0) \neq 0$. Si posem $a(x)$ com

$$a(x) = a_1(x)^{s_1} \cdots a_r(x)^{s_r}$$

amb $s_1 \leq \ldots \leq s_r$, on $a_i(x)$ són polinomis irreductibles d'ordre e_i, grau $n_i = \mathrm{ord}_{e_i}(2)$ i tals que $a_i(0) \neq 0$, aleshores, per les Proposicions 2.10 i 2.12, s'ha de complir

$$360 = 2^{\lceil s_r \rceil} \operatorname{mcm}\{e_1, \ldots, e_r\},$$
$$n = s_1 n_1 + \cdots + s_r n_r.$$

Com que $360 = 2^3 \cdot 3^2 \cdot 5$ i tots els e_i són coprimers amb 2 per ser ordres de polinomis irreductibles, necessàriament hem de tenir $\lceil s_r \rceil = 3$. Hi ha diversos s_r que compleixen això, però ja que com més gran prenguem s_r més gran serà n, prendrem s_r el més petit possible:

$$s_r = 2^{3-1} + 1 = 5.$$

Ara observem que la resta dels s_i per a $1 \leq i \leq r-1$ no apareixen a l'expressió de l'ordre de $a(x)$. A més, com més grans siguin més gran serà el grau de $a(x)$. Per tant prendrem

$$s_1 = s_2 = \ldots = s_{r-1} = 1.$$

Així doncs, hem restringit la cerca a polinomis de la forma

$$a(x) = a_1(x) \cdots a_{r-1}(x) a_r(x)^5.$$

A continuació explorarem les possibilitats que hi ha per a cada valor de r fins a trobar quin és el polinomi de grau mínim.

3.1.1 Cas $r = 1$

En aquest cas cal que tinguem $e_1 = 360/8 = 45$. Per tant, $a_1(x)$ ha de ser un polinomi irreductible d'ordre 45, que tindrà grau $n_1 = \mathrm{ord}_{45}(2) = 12$. Així

doncs, el polinomi $a(x)$ tindrà grau $n = 5 \cdot 12 = 60$.

3.1.2 Cas $r = 2$

En aquest cas hem de tenir un parell e_1, e_2 de manera que $\text{mcm}\{e_1, e_2\} = 45$. Això vol dir que tant e_1 com e_2 han de ser divisors de 45. A la Taula 3.1 hi tenim una llista dels divisors d de 45 juntament amb el grau n_d dels polinomi irreductibles que tenen per ordre aquest divisor.

d	n_d
1	1
3	2
5	4
9	6
15	4
45	12

Taula 3.1: Grau dels polinomis irreductibles que tenen per ordre algun divisor de 45

Òbviament, si d_1 i d_2 són una parella de divisors de 45 tals que $\text{mcm}\{d_1, d_2\} = 45$ i tenim $n_{d_1} \geq n_{d_2}$, aleshores tindrem

$$n_{d_2} + 5n_{d_1} \geq n_{d_1} + 5n_{d_2}.$$

Per tant, d'entre les dues possibilitats, agafarem $e_1 = d_1$ i $e_2 = d_2$ per tal d'obtenir un polinomi $a(x)$ de grau menor. L'assignació inversa $e_1 = d_2$ i $e_2 = d_1$ ens donaria un polinomi de grau major i ja no la considerarem.

Per altra banda, siguin d_1 i d_2 una parella de divisors de 45 tal que $45 = \text{mcm}\{d_1, d_2\}$ i d_3 un altre divisor de 45 de manera que $45 = \text{mcm}\{d_1, d_3\}$ i $n_{d_3} < n_{d_2}$. Aleshores està clar que prenent la parella d_1, d_3, obtindrem un polinomi de grau menor que prenent la parella d_1, d_2. Per tant tampoc considerarem les parelles d_1, d_2 per les quals passa el que acabem de descriure.

Després de totes aquestes consideracions, ens quedem només amb dues parelles possibles. La Taula 3.2 recull els graus dels polinomis que s'obtenen amb aquestes dues parelles.

e_1	e_2	n_1	n_2	n
45	1	12	1	17
9	5	6	4	26

Taula 3.2: Graus dels polinomis d'ordre 360 i grau mínim que podem obtenir amb $r = 2$

3.1.3 Cas $r = 3$

Repetint les mateixes consideracions que en el cas anterior podem eliminar moltes possibilitats. Per començar observem que 45 només té dos divisors primers diferents, 3 i 5. Per tant, si d_1, d_2, d_3 és una terna de divisors de 45 tals que $\text{mcm}\{d_1, d_2, d_3\} = 45$, aleshores 9 ha de dividir algun d_i i 5 ha de dividir algun d_j. Si $i = j$, aleshores tenim $d_i = 45$ i els altres dos divisors són redundants, o algun dels dos és 1 i el tercer és redundant. En ambdós casos podem obtenir graus menors prenent $r = 2$. Si $i \neq j$, aleshores $\text{mcm}\{d_i, d_j\} = 45$. Per tant aquesta parella ja la teniem en el cas $r = 2$. Suposem, sense pèrdua de generalitat, que $i = 1$, $j = 2$ i $n_{d_1} > n_{d_2}$. Aleshores prendriem $e_1 = d_1$ i $e_2 = d_2$. L'únic cas en el que podríem obtenir un polinomi $a(x)$ de grau menor afegint un tercer divisor d_3 amb polinomi irreductible de grau n_{d_3} seria quan

$$5n_{d_2} < n_{d_2} + 5n_{d_3} \Leftrightarrow 4n_{d_2} < 5n_{d_3}.$$

En el nostre cas les úniques possibilitats amb aquests requisits són $e_1 = 9$, $e_2 = 5$ i $e_3 = 3$ o $e_3 = 1$. Però el primer cas el podem descartar ja que el grau del polinomi irreductible d'ordre 3 és major que el d'ordre 1.

Per tant, l'únic cas que cal considerar és $e_1 = 9$, $e_2 = 5$ i $e_3 = 1$. En aquest cas tindriem un polinomi de grau

$$n = 6 + 4 + 5 = 15.$$

3.1.4 Cas $r \geq 4$

Totes les possibilitats que podem considerar a partir d'ara contindrien alguna de les que ja hem considerat fins ara i per tant ens donarien polinomis de grau major que els que ja hem obtingut. Per tant no seguim.

3.1.5 Conclusió

Els polinomis d'ordre 360 i grau mínim tenen grau 15 i són de la forma

$$a(x) = a_1(x)a_2(x)(x+1)^5$$

on $\mathrm{ord}(a_1(x)) = 9$ i $\mathrm{ord}(a_2(x)) = 5$. Usant el mètode de càlcul que hem explicat després de la Proposició 2.10 obtenim els següents polinomis:

$$\begin{aligned} a_1(x) &= x^6 + x^3 + 1, \\ a_2(x) &= x^4 + x^3 + x^2 + x + 1. \end{aligned}$$

Per tant tindrem

$$a(x) = x^{15} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1.$$

Usant aquest polinomi $a(x)$ i la llavor $u(x) = 1$ podem generar una $(2, 15, 360)$-seqüència tancada. A més, sabem que no és possible aconseguir una $(2, n, 360)$-seqüència tancada amb $\lceil \log_2 360 \rceil = 9 \leq n < 15$ usant una permutació lineal dels vèrtexs de $B(2, n)$ per les n esmentades.[1]

Per fer-ho hem de considerar el vector $u = (1, 0, \ldots, 0) \in \mathbb{F}_2^{15}$ associat al polinomi $a(x)$ i calcular la llavor $v = uP$ per a generar el cicle usant la matriu acompanyant de $a(x)$, on P és la matriu de la Proposició 2.6. Però en aquest cas el càlcul de v és immediat ja que es correspon amb la primera fila de P i aquesta no depén de $a(x)$. En definitiva, tenim $v = (0, \ldots, 0, 1) \in \mathbb{F}_2^{15}$. Ara hem de calcular la seqüència de vectors

$$v, vF, vF^2, \ldots, vF^{e-1},$$

on F és la matriu acompanyant de $a(x)$, i formar la seqüència que s'obté prenent la primera coordenada de cadascun d'aquests vectors. La seqüència que s'obté és la que es pot veure a la Figura 3.1.

[1] Vegeu els comentaris de la secció *Resultats experimentals* al respecte d'aquesta afirmació.

```
000000000000001001110100111100100101111001110
011101110111010100000011100001010010100100000
100110011001101111101101011010111100011111101
010001000100011000110000101101100001101000110
111111111111110110001011000011011010000110001
100010001000101011111100011110101101011011111
011001100110010000010010100101000011100000010
101110111011100111001111010010011110010111001
```

Figura 3.1: Una $(2, 15, 360)$-seqüència tancada.

3.2 El cas general

En aquesta secció volem generalitzar el raonament de la secció anterior al cas general per tal de poder donar un algorisme per resoldre el Problema 6.

Suposem que tenim un cos finit fixat $\mathbb{F}_q$ amb $q = p^k$. Ens donen un enter $e \geq 2$ i volem trobar un polinomi mònic de grau mínim $a(x) \in \mathbb{F}_q[X]$ de manera que $a(0) \neq 0$ i $\mathrm{ord}(a(x)) = e$. Si factoritzem e com a producte de potències de nombres primers diferents tindrem

$$e = p^{\alpha} p_1^{\alpha_1} \cdots p_t^{\alpha_t} = p^{\alpha} e_*,$$

amb $\alpha \geq 0$, $\alpha_i \geq 1$ i $p_i \neq p$ per a tot i, $1 \leq i \leq t$. Observem que e_* és coprimer amb p i per tant també amb q. A menys que diguem el contrari, a partir d'ara suposarem que tots els polinomis que apareixen són mònics i no són divisibles per x.

Hem vist a la secció anterior que si $a(x)$ és una solució del nostre problema, aleshores ha de ser de la forma

$$a(x) = a_1(x) \cdots a_{r-1}(x) a_r(x)^s$$

amb $s = p^{\alpha-1} + 1$ si $\alpha \geq 1$ i $s = 1$ si $\alpha = 0$. Els $a_i(x)$ són polinomis mònics irreductibles de $\mathbb{F}_q[X]$ d'ordre e_i i grau n_i per a tot i, $1 \leq i \leq r$. Recordem que si $e_i = 1$ aleshores $n_i = 1$ i $a_i(x) = x - 1$, altrament $n_i = \mathrm{ord}_{e_i}(q)$ que està ben definit perquè els $a_i(x)$ són irreductibles i per tant el seu ordre és coprimer amb q. Per tal de simplificar la notació redefinirem lleugerament l'operació ordre i direm que $\mathrm{ord}_b(a) = 1$ si $b = 1$ i altrament és el menor enter positiu i tal que $a^i \equiv 1 \mod b$. En aquest cas tindrem que l'ordre i el

grau de $a(x)$ seran

$$\begin{aligned}\operatorname{ord}(a(x)) &= p^{\alpha}\operatorname{mcm}\{e_1,\ldots,e_r\},\\ \deg(a(x)) &= n_1+\cdots+n_{r-1}+sn_r,\end{aligned}$$

amb $n_i = \operatorname{ord}_{e_i}(q)$. Observem que l'ordre en què apareixen els e_i en l'expressió de l'ordre de $a(x)$ no afecta el resultat, però l'ordre en què apareixen els n_i sí que afecta el grau de $a(x)$. Per tant, com que el nostre objectiu és aconseguir polinomis de grau mínim amb un ordre donat, suposarem, sense pèrdua de generalitat, que, en totes les expressions d'aquest tipus, $n_r = \min\{n_1,\ldots,n_r\}$. En cas contrari sempre podem canviar l'ordre dels factors per aconseguir un polinomi de grau menor i mateix ordre. Per tant, podem pensar que el nostre problema no té res a veure amb polinomis, sinó amb nombres coprimers amb q tals que el seu mínim comú múltiple compleix una certa propietat i que minimitzen una certa suma. Això motiva la següent definició.

Definició 3.1. Siguin q i e dos enters amb $q = p^k$ per algun primer p. Si denotem per e_* el major divisor de e coprimer amb q, aleshores tenim $e = p^{\alpha}e_*$ per algun $\alpha \geq 0$. Diem que un conjunt de r enters positius diferents $E = \{e_1,\ldots,e_r\}$ és un *conjunt d'ordres per a e respecte q* si es compleix que

1. e_i és coprimer amb q per a tot i, i
2. $\operatorname{mcm} E = e_*$.

Si E és un conjunt d'ordres per a e respecte q, posant $n_i = \operatorname{ord}_{e_i}(q)$ per a tot i, definim el seu *grau* com

$$n(E) = n_1+\cdots+n_{r-1}+n_r+(s-1)\min_{1\leq i\leq r}\{n_i\}$$

on $s = p^{\alpha-1}+1$ si $\alpha > 0$ i $s = 1$ si $\alpha = 0$. A més, direm que un conjunt $E = \{e_1,\ldots,e_r\}$ d'ordres per a e respecte q és *propi* si tenim $e_i \neq 1$ per a tot i, $1 \leq i \leq r$.

La primera observació respecte aquesta definició és que si E és un conjunt d'ordres per a e respecte q, aleshores podem suposar sense pèrdua de generalitat, i així ho farem quan ens convingui, que es compleix $n_1 \geq n_2 \geq \ldots \geq n_r$. En aquest cas la fórmula pel grau de E queda

$$n(E) = n_1+n_2+\cdots+n_{r-1}+sn_r.$$

La segona observació és que, fixats e i q, per a cada conjunt E d'ordres per a e respecte q podem trobar un polinomi $a_E(x) \in \mathbb{F}_q[X]$ d'ordre e i grau $n(E)$. Per fer-ho, prenem $a_i(x)$ un polinomi irreductible d'ordre e_i que sabem que tindrà grau n_i i agafem, suposant que els ordres estan ordenats per grau de manera decreixent,

$$a_E(x) = a_1(x) \cdots a_{r-1}(x) a_r(x)^s.$$

Per tant, buscar un polinomi a $\mathbb{F}_q[X]$ de grau mínim i ordre e és el mateix que buscar un conjunt d'ordres per a e respecte q de grau mínim. A partir d'ara abordarem aquest problema suposant que els enters q i e estan fixats. Per tant ens caldrà descobrir com són els conjunts d'ordres per a e respecte q que tenen grau mínim. És a dir, resoldre el problema següent.

Problema 7. Donats dos enters positius q i e amb $q = p^k$ per algun primer p, trobar de manera eficient un conjunt E d'ordres per a e respecte q de grau mínim.

La tercera observació és que per a cada conjunt d'ordres propi $E = \{e_1, \ldots, e_r\}$ tenim un únic conjunt d'ordres no propi $E' = \{e_1, \ldots, e_r, e_{r+1}\}$ amb $e_{r+1} = 1$ i viceversa. Observem que si els ordres de E estan ordenats per grau de manera decreixent, aleshores també ho estan els ordres de E'. El següent resultat ens diu quina relació que hi ha entre $n(E)$ i $n(E')$.

Lema 3.2. *Siguin $E = \{e_1, \ldots, e_r\}$ un conjunt d'ordres propi de grau $n(E)$ i $E' = \{e_1, \ldots, e_r, e_{r+1}\}$, $e_{r+1} = 1$, el seu conjunt d'ordres no propi associat amb grau $n(E')$. Aleshores, o bé $n(E') \leq n(E)$, o bé $n(E') = n(E) + 1$. A més, aquest segon cas només es dóna si $s = 1$ ó $n_r = 1$.*

Demostració. Suposem primer que $s = 1$. Aleshores tenim

$$\begin{aligned} n(E) &= n_1 + \cdots + n_{r-1} + n_r, \\ n(E') &= n_1 + \cdots + n_{r-1} + n_r + 1 \end{aligned}$$

i per tant $n(E') - n(E) = 1$. Ara suposem que $n_r = 1$. Aleshores tindrem

$$\begin{aligned} n(E) &= n_1 + \cdots + n_{r-1} + s, \\ n(E') &= n_1 + \cdots + n_{r-1} + 1 + s \end{aligned}$$

i per tant $n(E') - n(E) = 1$. Finalment, suposem que $s \geq 2$ i $n_r \geq 2$. En aquest cas hem de verificar que $n(E) - n(E') = sn_r - n_r - s \geq 0$, o,

equivalentment, que

$$n_r \frac{s-1}{s} \geq 1.$$

Però aquesta última desigualtat és certa ja que $n_r \geq 2$ i $\frac{s-1}{s} \geq \frac{1}{2}$ per ser $s \geq 2$. □

Per tant, donat un conjunt d'ordres propi E, podem decidir fàcilment si el seu conjunt d'ordres no propi associat, E', té grau menor.

Com que el nostre objectiu és buscar conjunts d'ordres de grau mínim i a priori hi ha molts conjunts d'ordres possibles, es fa convenient estudiar quines propietats tenen els conjunts d'ordres de grau mínim per tal de reduir el domini de la cerca. El següent és un primer resultat en aquesta línia.

Lema 3.3. *Sigui $E = \{e_1, \ldots, e_r\}$ un conjunt d'ordres per a e respecte q. Si per algun i, $1 \leq i \leq r$, amb $n_i \neq 1$, es té que* $\operatorname{mcm}(E \setminus \{e_i\}) = e_*$*, aleshores E no és de grau mínim.*

Demostració. Distingim dos casos. Si $i \neq r$ aleshores definim $E' = E \setminus \{e_i\}$ que, per hipòtesi, és un conjunt d'ordres per a e respecte q. Aleshores els graus de E i E' són

$$\begin{aligned} n(E) &= n_1 + \cdots + n_{i-1} + n_i + n_{i+1} + \cdots + n_{r-1} + sn_r, \\ n(E') &= n_1 + \cdots + n_{i-1} + n_{i+1} + \cdots + n_{r-1} + sn_r. \end{aligned}$$

Per tant $n(E) - n(E') = n_i > 0$ i E no és de grau mínim. Si $i = r$, definim E' com el conjunt que s'obté de E reemplaçant e_r per $e'_r = 1$. Com que $n'_r = 1$, aleshores aquest grau és menor o igual que la resta de graus i el podem col·locar en la útlima posició. Els graus de E i E' seran doncs

$$\begin{aligned} n(E) &= n_1 + \cdots + n_{r-1} + sn_r, \\ n(E') &= n_1 + \cdots + n_{r-1} + s. \end{aligned}$$

Per tant, utilitzant que $n_r > 1$, tenim

$$n(E) - n(E') = sn_r - s = s(n_r - 1) > 0,$$

i per tant $n(E') < n(E)$. □

Notem que si E conté un ordre redundant amb grau 1, aleshores el podem substituir per 1 sense alterar el grau del conjunt d'ordres. Per tant, el lema

ens diu que si estem buscant conjunts d'ordres de grau mínim, només cal que considerem els conjunts en els quals no hi ha ordres redundants llevat, potser, de l'1. I de fet, el Lema 2.14 ens permet provar un resultat més fort: per trobar conjunts d'ordres de grau mínim ens podem restringir a conjunts on tots els ordres són coprimers dos a dos.

Proposició 3.4. *Per a tot conjunt d'ordres per a e respecte q, $E = \{e_1, \ldots, e_r\}$, existeix un conjunt d'ordres $E' = \{e'_1, \ldots, e'_{r'}\}$ per a e respecte q amb $n(E') \leq n(E)$ tal que cadascun dels p_l de la factorització de e_* divideix un i només un dels e'_i.*

Demostració. Si a E cada p_l divideix un i només un dels e_i, aleshores ja hem acabat. Suposem que no, que existeix algun p_l que divideix e_i i e_j amb $i \neq j$. Com que $p_l^{\alpha_l}$ divideix e_*, aleshores ha de dividir algun dels ordres de E. Podem suposar que $p_l^{\alpha_l}$ divideix e_i i prenem β, $1 \leq \beta \leq \alpha_l$, el major enter positiu tal que p_l^{β} divideix e_j. Aleshores definim

$$e'_j = \frac{e_j}{{p_l}^{\beta}}.$$

Observem que e'_j divideix e_j i per tant, pel Lema 2.14, $\mathrm{ord}_{e'_j}(q)$ divideix $\mathrm{ord}_{e_j}(q)$ i tenim $\mathrm{ord}_{e'_j}(q) \leq \mathrm{ord}_{e_j}(q)$. Ara considerem les diferents possibilitats per e'_j.

D'una banda, si $e'_j = 1$ o e'_j és un ordre que ja es trobava en el conjunt E, aleshores e_j era redundant en el conjunt E. Per tant, pel Lema 3.3, eliminant e_j de E o reemplaçant-lo per 1 obtenim un conjunt d'ordres per a e respecte q, E', amb $n(E') \leq n(E)$.

Per altra banda, suposem que $e'_j \neq 1$ i no es troba en el conjunt E. Aleshores el conjunt E' que s'obté a partir de E reemplaçant e_j per e'_j és un conjunt d'ordres per a e respecte q. Si tenim $j = r$, aleshores tenim que $n(E') \leq n(E)$ ja que

$$n(E) - n(E') = s\,\mathrm{ord}_{e_j}(q) - s\,\mathrm{ord}_{e'_j}(q) \geq 0,$$

per ser $\mathrm{ord}_{e_j}(q) \geq \mathrm{ord}_{e'_j}(q)$. Si en canvi tenim $j \neq r$, aleshores podem distingir dos casos.

En primer lloc, si $n_r > \mathrm{ord}_{e'_j}(q)$ llavors en $n(E')$ el terme multiplicat per

s serà $\mathrm{ord}_{e'_j}(q)$ i per tant tindrem

$$\begin{aligned} n(E) - n(E') &= \mathrm{ord}_{e_j}(q) + sn_r - n_r - s\,\mathrm{ord}_{e'_j}(q) \\ &= s(n_r - \mathrm{ord}_{e'_j}(q)) + \mathrm{ord}_{e_j}(q) - n_r > 0, \end{aligned}$$

on hem utilitzat que $\mathrm{ord}_{e_j}(q) \geq n_r$ i $n_r > \mathrm{ord}_{e'_j}(q)$. Per tant $n(E') < n(E)$.

En segon lloc, si $n_r \leq \mathrm{ord}_{e'_j}(q)$ llavors tenim

$$n(E) - n(E') = \mathrm{ord}_{e_j}(q) - \mathrm{ord}_{e'_j}(q) \geq 0.$$

És a dir, $n(E') \leq n(E)$.

Finalment, veiem que en tots els casos podem obtenir un conjunt d'ordres E' per a e respecte q amb $n(E') \leq n(E)$ de manera que p_l divideix un ordre menys que en E. Si E' no és de la forma desitjada, alsehores podem iterar aquest procés, obtenint en cada iteració un nou conjunt d'ordres amb grau menor o igual que l'anterior fins a arribar a un conjunt d'ordres de la forma desitjada. □

En vistes d'aquest últim resultat ja podem esbossar una estratègia a seguir per un algorisme que resolgui el Problema 7. Donats $q = p^k$ i

$$e = p^{\alpha} p_1^{\alpha_1} \cdots p_t^{\alpha_t},$$

explorem tots els conjunts d'ordres propis per e respecte q de la forma

$$E = \{e_1, \ldots, e_r\}$$

on cada e_i és el producte d'un o més nombres de la forma $p_l^{\alpha_l}$ de manera que cada primer p_l apareix en la factorització de un i només un dels e_i. Per cadascun d'aquests conjunts E, comprovem si $n(E') \leq n(E)$ on E' és el conjunt d'ordres no propi associat a E. En cas afirmatiu ens quedem amb E', altrament ens quedem amb E. Un cop hagim acabat tindrem una col·lecció finita de conjunts d'ordres per a e respecte q entre els quals n'hi ha, com a mínim, un de grau mínim. Aquesta és l'estratègia que utilitza l'algorisme que presentem en la pròxima secció.

Per acabar aquesta secció formalitzarem un fet que està implícit en l'estratègia que acabem d'exposar i que usarem per a la construcció de l'algorisme que busca un conjunt d'ordres de grau mínim per a e respecte q.

Corol·lari 3.5. *La col·lecció formada per tots el conjunts d'ordres E per a e respecte q en els quals cada primer p_j que apareix en la factorització de e_* divideix un i només un dels ordres de E està en bijecció amb*

$$\mathcal{P}\left(\{p_1^{\alpha_1}, \ldots, p_t^{\alpha_t}\}\right),$$

la col·lecció de totes les particions del conjunt format pels divisors primers diferents de p que apareixen en la factorització de e_* elevats als seus respectius exponents.

3.3 L'algorisme

En aquesta secció presentem el pseudo-codi d'un algorisme que resol el Problema 6. Una versió d'aquest algorisme apareixerà publicat a Fuertes *et al.* (2008). Donats dos enters, $q = p^k$ per algun primer p i $e = p^{\alpha} p_1^{\alpha_1} \cdots p_t^{\alpha_t}$, l'algorisme troba un polinomi $a(x) \in \mathbb{F}_q[X]$ d'ordre e i grau mínim. L'algorisme es basa en l'estratègia que hem esbossat en la secció anterior i està dividit en dues parts.

La primera part és l'Algorisme 1 que implementa el procediment que hem anomenat **EGrauMinim**. Aquest procediment, donats els enters q i e factoritzats tal com s'indica, resol el Problema 7 fent us de les idees que hem exposat en la secció anterior.

La segona part és l'Algorisme 2 que resol el problema en sí. Aquest algorisme usa el procediment **EGrauMin** per determinar quina és la millor manera d'agrupar els divisors primers de e diferents de p per tal d'aconseguir un polinomi de grau mínim. A partir d'aquí obté una llista d'ordres i per cadascun d'aquests ordres computa un polinomi irreductible de $\mathbb{F}_q[X]$ d'aquest ordre. Finalment multiplica tots els polinomis i eleva a la potència s el que té grau mínim. D'aquesta manera obté un polinomi $a(x)$ d'ordre e i grau mínim.

3.4 Anàlisi de la complexitat

Tot i que un anàlisi detallat de la complexitat de l'algorisme presentat en la secció anterior s'escapa de l'abast d'aquest treball, en aquesta secció justificarem perquè diem que l'algorisme que hem presentat és eficient.

Input: Dos enters factoritzats, $q = p^k$ i $e = p^\alpha p_1^{\alpha_1} \cdots p_t^{\alpha_t}$
Output: Un conjunt d'ordres per a e respecte q de grau mínim
begin
  if $t = 0$ **then return** $\{1\}$
  if $\alpha = 0$ **then** $s \leftarrow 1$ **else** $s \leftarrow p^{\alpha-1} + 1$
  $D \leftarrow \{p_1^{\alpha_1}, \ldots, p_t^{\alpha_t}\}$
  $n_{min} \leftarrow \infty$
  $\mathcal{S}_{min} \leftarrow \emptyset$
  foreach $\mathcal{S} \in \mathcal{P}(D)$ **do** /* $\mathcal{S} = \{P_1, \ldots, P_r\}$ */
    $m \leftarrow \infty$ /* $D = P_1 \cup \ldots \cup P_r$ */
    foreach $P_i \in \mathcal{S}$ **do**
      $n_i \leftarrow \operatorname{mcm}_{d \in P_i} \{\operatorname{ord}_d(q)\}$
      if $n_i < m$ **then** $m \leftarrow n_i$
    end
    if $m = 1 \vee s = 1$ **then**
      $n \leftarrow \sum_{i=1}^{r} n_i + (s-1)m$
    else
      $n \leftarrow \sum_{i=1}^{r} n_i + s$
    end
    if $n < n_{min}$ **then**
      $n_{min} \leftarrow n$
      $\mathcal{S}_{min} \leftarrow \mathcal{S}$
    end
  end
  $E \leftarrow \emptyset$
  foreach $P_i \in \mathcal{S}_{min}$ **do**
    $e_i \leftarrow \operatorname{mcm} P_i$
    $E \leftarrow E \cup \{e_i\}$
  end
  return E
end

Algorisme 1: Procediment **EGrauMinim**

Input: Dos enters positius, $q = p^k$ i $e \geq 2$
Output: Un polinomi $a(x) \in \mathbb{F}_q[X]$ d'ordre e i grau mínim
begin
  $(p, k) \leftarrow$ **Factoritza**(q)
  $(\alpha, t, p_1, \alpha_1, \ldots, p_t, \alpha_t) \leftarrow$ **Factoritza**(e)
  if $\alpha = 0$ **then** $s \leftarrow 1$ **else** $s \leftarrow p^{\alpha-1} + 1$
  $E \leftarrow$ **EGrauMinim**$(q = p^k, e = p^\alpha p_1^{\alpha_1} \cdots p_t^{\alpha_t})$
  $a(x) \leftarrow 1$
  $n_{min} \leftarrow \infty$
  $a_{min}(x) \leftarrow 1$
  foreach $e_i \in E$ **do**
    $a_i(x) \leftarrow$ un polinomi irreductible de $\mathbb{F}_q[X]$ d'ordre e_i
    $n_i \leftarrow \deg(a_i(x))$
    $a(x) \leftarrow a(x)a_i(x)$
    if $n_i < n_{min}$ **then**
      $n_{min} \leftarrow n_i$
      $a_{min}(x) \leftarrow a_i(x)$
    end
  end
  $a(x) \leftarrow a(x)a_{min}(x)^{s-1}$
  return $a(x)$
end

Algorisme 2: Solució del Problema 6

El primer pas és estudiar la complexitat de l'Algorisme 1. Suposem que tenim uns paràmetres d'entrada donats, $q = p^k$ i $e = p^\alpha p_1^{\alpha_1} \cdots p_t^{\alpha_t}$. Òbviament, el gros de la computació de **EGrauMin**(q, e) se l'emporta el bucle que recorre totes les possibles particions del conjunt $D = \{p_1^{\alpha_1}, \ldots, p_t^{\alpha_t}\}$. El nombre d'iteracions d'aquest bucle vé donat per $B_t = |\mathcal{P}(D)|$, on B_t és el t-èsim nombre de Bell que calcula el nombre de possibles particions d'un conjunt de t elements. Els nou primers nombres de Bell són

$$B_0 = 1,\ B_1 = 1,\ B_2 = 2,\ B_3 = 15,\ B_4 = 52,$$
$$B_5 = 203,\ B_6 = 877,\ B_7 = 4140 \text{ i } B_8 = 211747.$$

Com veiem, aquests nombres creixen molt ràpid amb t i per tant si el nombre de factors primers diferents de e és gran aleshores el bucle haurà de realitzar moltes passades.

Per poder estimar el nombre d'iteracions del bucle principal en funció del nombre de divisors primers de e diferents de p, donarem una cota superior pels nombres de Bell. La cota que donarem no és la millor que es coneix, però té una expressió analítica senzilla i ens permetrà fer càlculs aproximats per justificar l'eficiència de l'Algorisme 2.

La següent és una expressió per l'enèsim nombre de Bell en funció dels anteriors nombres de Bell que es pot trobar en molts textes introductoris de combinatòria. Vegeu, per exemple, Cameron (1995).

$$B_n = \sum_{k=0}^{n-1} \binom{n-1}{k} B_k$$

Usant que els nombres de Bell són creixents amb n i la identitat combinatòria

$$\sum_{k=0}^{n-1} \binom{n-1}{k} = 2^{n-1},$$

tenim que

$$B_n \leq B_{n-1} 2^{n-1}.$$

per a tot $n \geq 1$. Això ens permet provar per inducció sobre n la següent cota superior pels nombres de Bell:

$$B_n \leq 2^{\frac{n(n-1)}{2}} \leq 2^{n^2},$$

per a tot $n \geq 0$.

Concloem doncs, que el nombre d'iteracions del bucle principal de l'Algorisme 1 és, com a molt, exponencial en t^2, on t és el nombre de divisors primers de e diferents de p.

Ara analitzarem la complexitat total de la solució, la de l'Algorisme 2. A part de la crida al procediment **EGrauMinim**, les operacions rellevants que l'algorisme realitza són basicament dues:

1. Factorització d'enters, i
2. Multiplicació i divisió de polinomis de $\mathbb{F}_q[X]$.

Els detalls dels algorismes que realitzen aquestes operacions no són rellevants pel nostre algorisme, però sí la seva complexitat. Com que existeixen múltiples algorismes per resoldre aquests problemes, vegeu per exemple VON ZUR GATHEN i GERHARD (1999), la complexitat exacte d'aquests càlculs dependrà dels detalls de l'implementació en qüestió. No obstant, pels nostres propòsits és suficient saber que existeixen algorismes que resolen aquests problemes de manera eficient quan les entrades són de mida raonable.

Més explícitament, la factorització d'un enter N es pot fer en un nombre d'operacions proporcional a $N^{\frac{1}{4}}(\log N^{\frac{1}{4}})^c$ per a una certa constant positiva c. Tot i que per a valors grans de N aquesta complexitat es torna impracticable, en el nostre cas els nombres a factoritzar, q i e, són relativament petits respecte als valors que s'usen en criptografia a dia d'avui. Concretament, es consideren difícils de factoritzar usant un ordinador estàndard nombres de 512 o més bits. Com que $2^{512} \approx 10^{154}$, estem parlant de nombres molt més grans que 10^{80}, nombre que es considera una estimació a la baixa del nombre d'àtoms de materia visible a l'Univers. En qualsevol aplicació d'enginyeria on es vulgui construir una (m, n, e)-seqüència per "escriure-la" en un dispositiu s'usarà un nombre d'àtoms molt més petit que 10^{80} i el nombre e haurà de ser encara més petit que 10^{80}. Per tant podem dir que per als valors típics de q i e, la seva factorització no representa una dificultat.

Per altra banda, la multiplicació i divisió de polinomis es pot realitzar en un nombre d'operacions polinòmic en el grau del polinomi més gran implicat. Com que els polinomis que apareixen a l'algorisme tenen per ordre un divisor de e, aleshores el seu grau estarà acotat per sobre per e i aquestes operacions resultaran polinòmiques en e. La factorització de polinomis en un cos finit $\mathbb{F}_q$ també es pot fer de manera eficient, concretament amb un nombre d'operacions polinòmic en el grau i polinòmic en $\log q$. A més, aquestes

operacions es realitzaran, com a molt, un nombre polinòmic de vegades amb els paràmetres d'entrada.

Per tant, tots els càlculs que apareixen en l'Algorisme 2 es poden fer amb un nombre d'operacions que és polinòmic en els paràmetres d'entrada llevat de la crida al procediment **EGrauMinim**. El nombre d'operacions que realitza aquest procediment, com ja hem vist, és exponencial en t^2, on t és el nombre de factors primers diferents de p que apereixen en la factorització de e. Per tal de determinar quina és la complexitat mitja total de l'algorisme usarem el següent resultat de la teoria analítica de nombres que es pot trobar a Apostol (1998).

Proposició 3.6. *Sigui n un enter positiu i $\omega(n)$ el nombre de divisors primers diferents que apareixen en la factorització de n. Aleshores el comportament asimptòtic de l'esperança matemàtica de $\omega(n)$ és*

$$\mathbb{E}[\omega(n)] \approx \log(\log n).$$

Això, juntament amb la cota superior per B_t que hem deduït, ens està dient que una cota superior per a la mitjana del nombre d'operacions necessàries per evaluar **EGrauMinim**(q, e) és

$$2^{(\log(\log e))^2} = (\log e)^{\log 2 \log(\log e)}.$$

Com que es compleix que

$$\lim_{e\to\infty} \frac{(\log e)^{\log 2 \log(\log e)}}{e} = 0,$$

aleshores podem dir que el nombre esperat d'operacions necessàries per buscar el conjunt d'ordres per a e respecte q de grau mínim és, com a molt, polinòmic en e.

Per tant hem vist que el nombre esperat d'operacions que efectua l'Algorisme 2 amb entrades e i q és polinòmic en aquests dos nombres. Això ens permet concloure que el nostre algorisme és molt més eficient que la cerca per força bruta, que, tal com hem vist a 2.2, requereix un nombre esperat d'operacions exponencial en e.

Finalment, volem remarcar que, tot i que l'Algorisme 2 és molt més eficient que la cerca per força bruta en termes relatius, no és eficient en termes absoluts. La Complexitat Computacional és la disciplina que estudia el com-

portament asimptòtic del nombre d'operacions que realitza un algorisme en funció de la *mida* de la seva entrada. En el marc d'aquesta teoria, quan un algorisme té com entrada un o més nombres naturals, es considera que la mida d'aquesta entrada és el nombre de bits necessaris per representar aquests enters. Per a un enter x aquest nombre es comporta asimptòticament com $\log_2 x$. Tot i que no es pot considerar com una definició formal, s'acostuma a considerar que un algorisme és eficient quan el nombre d'operacions que realitza és polinòmic en la mida de l'entrada. En el nostre cas el nombre d'operacions és polinòmic en l'*entrada*, no la seva mida. De fet, en aquests termes, l'algorisme és exponencial en la mida de l'entrada. No obstant, convé observar que el simple fet d'*escriure* un seqüència de longitud e requereix executar e operacions, que és un nombre exponencial en $\log_2 e$. Per tant, tot i que teòricament fos possible trobar un polinomi per generar una (q, n, e)-seqüència en temps polinòmic en la mida de les entrades q i e, en última instància, per obtenir la seqüència es requereix un nombre d'operacions exponencial en la mida de l'entrada.

3.5 Resultats experimentals

En aquesta secció comentarem un fenomen que hem observat en experimentar amb l'algorisme de la secció 3.3. Abans però farem dues observacions rellevants de cara al fenomen estudiat.

En primer lloc, és un fet bastant obvi que si tenim una (m, n, e) seqüència tancada x, aleshores x és també una (m', n', e)-seqüència per a qualssevol $m' \geq m$ i $n' \geq n$. Clarament si afegim símbols a una alfabet de m símbols fins a obtenir-ne un de m' símbols, la seqüència manté la seva longitud, les subparaules de longitud n continuen essent totes diferents i per tant el canvi d'alfabet no té cap efecte. Més interessant és el cas en qual deixem m fix i incrementem n. En aquest cas, si les subparaules de x de longitud n són totes diferents, aleshores també ho són les subparaules de longitud $n' \geq n$. Per altra banda, això ens està dient que una porció de totes les (m, n', e)-seqüències tancades són també (m, n, e)-seqüències tancades per algun $n < n'$.

En segon lloc, considerem un polinomi mònic $a(x) \in \mathbb{F}_q[X]$, $a(0) \neq 0$, de grau n i ordre e i estructura cíclica $\mathcal{CS}(a(x))$. Aleshores, per cada element de l'estructura cíclica existeix, com a mínim, una llavor $u(x) \in \mathbb{F}_q[X]$ de grau menor que n que genera un cicle d'aquesta longitud. En el cas de l'ordre de

$a(x)$ podem prendre llavor $u(x) = 1$. I de fet qualsevol polinomi que aparegui en el cicle generat per $u(x) = 1$ també serà una llavor d'aquest cicle. Però a més, en general, existeixen altres cicles de longitud e en la descomposició cíclica de $\mathbb{F}_q[X]/(a(x))$ induïda per la bijecció $u(x) \mapsto xu(x)$. El nombre total de cicles de longitud e en aquesta descomposició cíclica es pot comptar tot i que en general no surt una fórmula senzilla de manejar (VENTURA, 1997).

Vistes aquestes dues observacions, és natural plantejar-se la següent pregunta. És possible que alguna de les (q, n, e)-seqüències tancades que podem obtenir de la descomposició cíclica de $\mathbb{F}_q[X]/(a(x))$ induïda per $a(x)$ sigui una (q, n', e)-seqüència tancada amb $n' < n$? En el cas que la resposta fos afirmativa, aquesta seqüència, obtinguda utilitzant un polinomi de grau n, ens permetria construir un codificador angular uni-corona de resolució e amb n' detectors. Per tant estariem disminuint el nombre de detectors necessaris en la contrucció del codificador.

En general sembla difícil estudiar quina pot ser aquesta n' i decidir si es pot aconseguir $n' = \lceil \log_q e \rceil$, el mínim teòric. No obstant, un cop tenim $a(x)$ és relativament senzill escriure una algorisme que calculi totes les (q, n, e)-seqüències tancades que es poden generar a partir de $a(x)$ i obtingui, per cadascuna, el mínim n' pel qual són (q, n', e)-seqüències tancades. Això és que hem fet amb el polinomi d'ordre 360 obtingut a 3.1. Els resultats es poden veure a la Taula 3.3. La primera fila de la taula compta el nombre k de seqüències per cadascun dels n' possibles. La tercera fila reflexa el total de detectors estalviats en cadascun dels casos. La taula mostra com en totes les seqüències podem reduir almenys un detector. Els resultats també mostren que en aquest cas és possible estalviar un màxim de 4 detectors i que per tant no s'arriba al mínim teòric de $n' = \lceil \log_2 360 \rceil = 9$.

Aquests resultats fan pensar que en la majoria de casos podrem estalviar alguns detectors respecte el nombre teòric necessari n. No obstant, és difícil saber, a priori, quants detectors serà possible estalviar.

k	–	16	12	6	8	–	–
n'	15	14	13	12	11	10	9
$15 - n'$	0	1	2	3	4	5	6

Taula 3.3: Relació de detectors estalviats

Capítol 4

Altres aplicacions i treballs futurs

En aquest capítol farem dues coses. D'una banda, presentarem una aplicació d'enginyeria que es pot beneficiar del nostre algorisme. Aquesta és el disseny de sistemes de localització per a vehicles guiats automàticament que operen en magatzems i plantes de manufactura. Veurem com els valors típics dels paràmetres de les (m, n, e)-seqüències tancades necessàries en aquest context fan que el nostre algorisme sigui particularment útil.

Per altra banda, presentarem una sèrie de reflexions sobre treballs futurs als quals s'obre la porta a la vista dels resultats exposats en aquesta memòria. En els punts que sigui possible, senyalarem algun element bibliogràfic que pugui servir de punt de partida per aquests treballs.

4.1 Vehicles guiats automàticament

En aquesta secció comentarem una altra possible aplicació de les (m, n, e)-seqüències tancades que es pot beneficiar del nostre algorisme.

Un *vehicle guiat automàticament*, en endavant VGA, és un robot mòbil emprat per moure materials en una planta de manufactura o en un magatzem. Tal com el seu nom indica, aquests vehicles es mouen de manera automàtica sense la necessitat de cap operari que els guiï.

Per tal que el VGA sàpiga quin recorregut ha de fer per dins d'un recinte tancat existeixen diversos mètodes de guiatge que es divideixen, bàsicament, en dos tipus.

D'una banda, hi ha els mètodes que es basen en marcar físicament el camí que ha de seguir el VGA al terra o a la paret del recinte. Per fer-ho es poden usar, des d'una línia pintada al terra que el robot segueix mitjançant un sensor òptic, fins a un cable enterrat que genera camps electromagnètics que el robot detecta i segueix. En qualsevol d'aquests mètodes, el robot és capaç de seguir el camí marcat però no té manera de saber en quin punt del recorregut es troba. Les parades i les bifurcacions s'han de marcar en el recorregut mitjançant algun tipus de marques addicionals.

Per altra banda, tenim els mètodes basats en la detecció per part del VGA de la seva posició mitjançant, per exemple, blancs reflectors de làser. En aquest cas el robot està equipat amb un emissor i un receptor de làser. Usant aquest dispositius, el robot localitza els blancs reflectors que hi ha en el recinte i mesura la distància que el separa de cadascun d'ells. Aleshores, mitjaçant un mapa del recinte que guarda a la memòria, calcula la seva posició i la utilitza per seguir sobre el mapa una ruta que prèviament se li ha programat.

Òbviament, el segon sistema és molt més flexible que el primer, però alhora és més costós, tant pel què fa a la maquinària del robot com a la complexitat de la programació i configuració del sistema. Per tal d'arribar a un compromís entre els dos enfocaments, l'article Petriu (1998) presenta un sistema que emprant m-seqüències tancades permet al VGA conèixer en tot moment la seva posició al llarg del recorregut sense la necessitat d'un sistema de posicionament tant complex com el que hem comentat.

El sistema és una modificació sobre el seguiment d'un línia pintada al terra. Consisteix en dividir la línia en petits segments d'una longitud fixa i etiquetar cadascun dels segments mitjançant una de m marques possibles. Si el recorregut queda dividir en un total de e segments, diposant les marques dels segments de manera que formin una (m, n, e)-seqüència i equipant el robot amb un dispositiu per llegir el símbol que hi ha en cada segment que recorre, aquest és capaç de conèixer en tot moment en quina posició del recorregut es troba. A la Figura 4.1 podem veure un esquema d'aquest muntatge en un magatzem on el VGA pot recórrer un circuit que li dóna accés a tots els prestatges on hi ha el material emmagatzemat i carregar-lo fins al moll de càrrega on arriben els camions.

Quan es considera una possible implementació d'aquest sistema hi ha dos factors que cal tenir en compte respecte els paràmetres de la seqüència.

D'una banda, que les mides del VGA són fixes mentre que el paràmetre

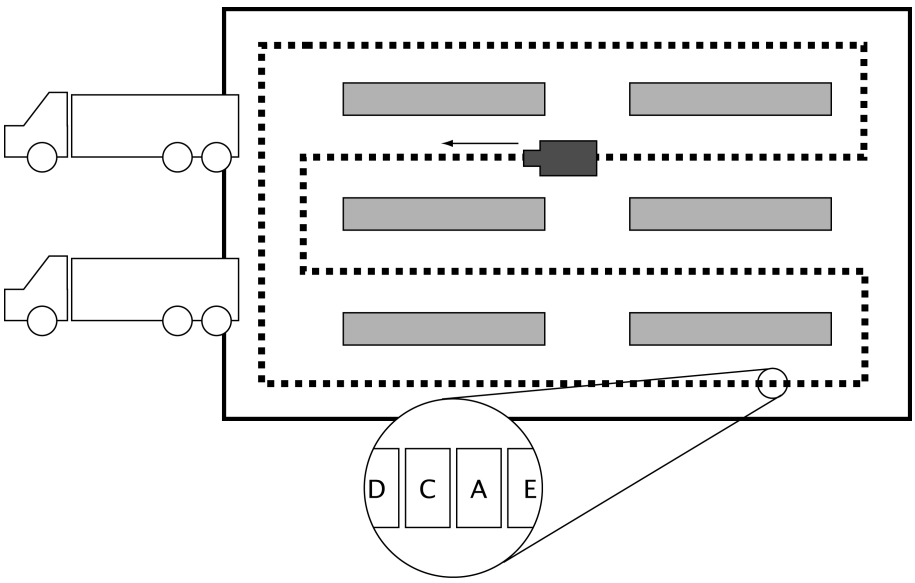

Figura 4.1: Magatzem amb un VGA que recorre un circuit marcat amb una seqüència tancada.

n de la seqüència varia amb e. Per tant no podem esperar que el robot sigui capaç de llegir n símbols de cop, sigui quin sigui n. Això implica que el robot no pot saber immediatament la seva posició en engegar-lo, necessita recórrer l'espai corresponent a n marques per tal de llegir-les i descobrir la seva posició. A la pràctica, però, aquest espai serà molt petit comparat amb la longitud total del recorregut i aquest efecte no representarà cap inconvenient substancial.

Per altra banda, com que, fixat e, n decreix a mesura que creix m, és convenient prendre m el més gran possible per reduïr la longitud que ha de recórrer el robot en un principi per tal de conèixer la seva posició. També és important reduïr aquesta longitud perquè, en el cas d'un error de lectura en una marca, és la longitud que ha de recórrer el robot per tal de recuperar-se de l'error de lectura i obtenir altre cop la posició correcte. L'ús de dispositius electrònics senzills, com els que interessa fer servir per construir codificadors angulars, permet distingir entre un nombre molt reduït de símbols diferents. Per aquesta raó es fa necessari l'ús d'altres tècniques per a l'identificació i reconeixement dels símbols. Seguint aquesta línia de pensament, els autors de Khalfallah *et al.* (1992) proposen usar sistemes de visió per ordinador per dur a terme aquesta tasca. És a dir, equipar el robot amb una càmara

que capti les imatges de la línia del terra a mesura que aquesta és resseguida pel robot i mitjançant un sistema de processament de la imatge identifiqui els símbols corresponents.

Com que en aquesta aplicació els valors típics de e són grans, per exemple $e = 10000$ per a un recorregut de 500 m amb marques de 5 cm, el cost exponencial de la cerca per força bruta el converteix en un mètode inviable pel disseny de les seqüències en qüestió. Recordem que la complexitat de la cerca per força bruta, a part de ser exponencial en e, també és creix amb m. Per tant, al prendre m gran també estem incrementant la complexitat de la cerca. Per altra banda, l'Algorisme 2 té complexitat polinòmica en e i en q, on q fa el paper de m però només pot prendre valors que siguin de la forma $q = p^k$ amb p primer. Disposant d'un sistema de reconeixement de símbols basat en visió per ordinador, aquesta restricció presenta un problema d'ordre menor. A més, la millora en la dificultat per obtenir la seqüència és notable.

Remarquem també que tot i que les seqüències tancades les hem pensat fins ara com un mètode per marcar les posicions d'un recorregut circular, sense principi ni final, també les podem usar per marcar un recorregut lineal, amb principi i final. Si tenim una (m, n, e)-seqüència tancada, $x = (x_0, x_1, \ldots, x_{e-1})$, disposada de manera lineal, aleshores movent-nos per sobre la seqüència podem llegir les $e-n+1$ paraules $x^j = (x_j, x_{j+1}, \ldots, x_{j+n-1})$ per a tot j, $0 \leq j \leq e - n$, que seran totes elles diferents per construcció. Això ens diu que els recorreguts d'un VGA que usi m-seqüències per conèixer la seva posició no estan restringits només a recorreguts circulars.

4.2 Possibles treballs futurs

En aquesta secció comentarem algunes possibilitats per a futurs treballs de recerca que els resultats exposats en aquesta memòria suggereixen.

En primer lloc, l'extensió de les tècniques presentades per a la generació de (m, n, e)-seqüències tancades amb m un enter qualsevol. En aquest cas, enlloc d'identificar l'alfabet sobre el qual estan definides les seqüències amb un cos finit, ho podem fer amb un anell finit, $\mathbb{Z}/m\mathbb{Z}$. Aleshores enlloc de parlar d'espais vectorials haurem de parlar de mòduls. S'hauria d'estudiar aleshores, usant la teoria de mòduls, fins a quin punt la teoria desenvolupada al Capítol 2 es pot generalitzar en aquest nou context. En aquesta línia es poden trobar alguns resultats a Bollman (1965).

En segon lloc, anant una mica més enllà que el punt anterior, l'ús de

tècniques no lineals per a la generació de seqüències tancades. Per la facilitat que presenta el seu estudi, en el Capítol 2 ens hem restringit a considerar bijeccions lineals del conjunt $\mathbb{F}_q^n$ en ell mateix de la forma

$$F(x_1, \ldots, x_n) = (x_2, \ldots, x_n, f(x_1, \ldots, x_n))$$

on f és una funció lineal dels $x_1, \ldots, x_n$ en la qual x_1 està ponderat per un coeficient no-zero. La idea consisteix en reemplaçar la funció lineal f per una funció no lineal dels $x_1, \ldots, x_n$ a partir del valor de la qual sigui possible recuperar x_1 tinguent els valors de $x_2, \ldots, x_n$. Aquesta condició garantiria la bijectivitat de F. El cas en què f és una funció afí dels $x_1, \ldots, x_n$, és a dir

$$f(x_1, \ldots, x_n) = a_1 x_1 + \cdots + a_n x_n + a,$$

ja ha estat estudiat des del punt de vista del període màxim, que nosaltres hem anomenat ordre (Lidl i Niederreiter, 1997). En el cas que f sigui un polinomi de grau més gran que 1 en n variables sembla que l'estudi serà més difícil. Des del punt de vista de teoria de nombres i de sistemes dinàmics existeixen gran quantitat de resultats en el cas $n = 1$ (Poorten *et al.*, 2003).

En tercer lloc, és possible pensar que les propietats combinatòries dels grafs de deBruijn puguin donar lloc a un algorisme per construir cicles de longitud donada sobre aquests grafs sense necessitat d'emprar tècniques algebraiques. En aquest cas caldria fer ús a la Teoria de Grafs per estudiar les propietats particulars d'aquests grafs. En aquest marc també es podria estudiar la complexitat intrínseca del problema d'obtenir un cicle de longitud arbitrària en un graf d'aquesta família. Això permetria classificar el problema en alguna de les classes de complexitat que s'usen en la teoria de la Complexitat Computacional. Aquesta classificació tindria com a conseqüència l'obtenció d'una cota inferior per a la complexitat de qualsevol algorisme que resolgués el problema.

En quart lloc, estudiar la possibilitat d'usar la redundància que introdueixen el mètodes lineals emprats per resoldre el problema per a la correcció d'errors. Donats q i e el nostre algorisme ens permet obtenir una (m, n, e)-seqüència tancada amb $n \geq \lceil \log_q e \rceil$. Quan aquesta n és més gran que $\lceil \log_q e \rceil$ aleshores estem usant més símbols dels necessaris per marcar cadascuna de les e posicions. Tal com hem assenyalat en 3.5, hi ha casos en què les e paraules de longitud n que conformen la seqüència també són diferents si prenem una finestra de mida inferior a n. Convindria estudiar si, enlloc de

suprimir aquesta redundància com hem suggerit anteriorment, la podem fer servir per corregir possibles errors de lectura en els detectors. En aquesta línia, el primer pas seria estudiar la distància de Hamming dels codis que s'obtenen per tal d'estimar el seu poder corrector.

Finalment, des d'un punt de vista més aplicat, buscar més problemes provinents del camp de l'enginyeria en els quals la possibilitat d'obtenir (q, n, e)-seqüències tancades de longitud arbitrària de manera eficient comporti una millora o una ajuda per a la seva resolució.

Capítol 5

Conclusions

A continuació resumim les metes assolides, els resultats obtinguts i les conclusions a les quals hem arribat en aquesta memòria.

En primer lloc, en el Capítol 1 hem vist en què consisteix la codificació angular i hem presentat les diferents famílies de codificadors angulars que existeixen. Hem vist que en els codificadors incrementals els errors presenten un comportament acumulatiu que no és desitjable en segons quines aplicacions. Això ens ha dut a considerar els codificadors absoluts en les seves diverses variants. Hem vist que l'ús d'una sola corona estalvia espai en la construcció del codificador i que aquesta propietat fa que els codificadors absoluts uni-corona siguin especialment útils per a aplicacions que requereixen una resolució elevada. El disseny de codificadors uni-corona usa seqüències generades mitjançant circuits LFSR i això comporta una sèrie de restriccions. La generalització d'aquestes seqüències i la seva construcció es planteja com l'objectiu a assolir per tal de poder construir codificadors uni-corona de resolució arbitrària.

Seguidament, en el Capítol 2 hem plantejat aquest problema en un llenguatge estrictament matemàtic. Aquest plantejament ens ha dut a considerar el que hem anomenat (m, n, e)-seqüències tancades. Hem justificat que aquest objecte és la generalització adequada en el nostre context de les seqüències generades per circuits LFSR. Després de veure que no és possible construir de manera eficient aquestes seqüències mitjançant la cerca per força bruta hem decidit buscar altres mètodes. Passant pels digrafs de de-Bruijn hem vist com usar mètodes algebraics per a la construcció d'aquestes seqüències. L'estudi d'aquests mètodes ens ha portat a considerar un conjunt de valors particulars pels paràmetres de la seqüència. Per aquests valors ens

ha estat possible emprar la teoria de cossos finits i l'àlgebra lineal per obtenir de manera eficient bones aproximacions a la solució del problema inicial.

En tercer lloc, en el Capítol 3 hem construit un algorisme per obtenir aquestes aproximacions usant la teoria desenvolupada en el capítol anterior. Hem demostrat que aquest algorisme obté la millor aproximació possible usant les tècniques considerades. Estudiant la complexitat de l'algorisme s'ha vist que aquest és eficient, com a mínim comparativament amb la cerca bruta. Així doncs l'algorisme obtingut representa la solució al problema que ens haviem marcat inicialment. El capítol acaba amb uns resultats obtinguts de manera experimental usant l'algorisme. Aquest resultats ens permet veure que, tot i que el nombre de detectors teòricament necessaris per contruir un codificadors angular uni-corona usant les seqüències que podem obtenir mitjançant l'algorisme sigui un, aquest nombre és en realitat menor en la majoria dels casos permetent l'estalvi d'alguns dels detectors.

Finalment, en el Capítol 4 hem considerat l'aplicació del nostre algorisme al problema de la localització de vehicle guiats automàticament. L'eficiència de l'algorisme el fa especialment útil en aquest context degut a què els paràmetres típics són elevats i la cerca per força bruta es torna impracticable. Per acabar hem comentat diverses línies de treball futur que s'obren a la vista dels resultats exposats i hem donat bibliografia relacionada amb aquests temes.

En resum, hem plantejat un problema en el domini de l'enginyeria, el del disseny de codificadors angulars uni-corona de resolució arbitrària. Hem abstret les propietats d'una bona solució i les hem traduït al llenguatge matemàtic. Usant mètodes de l'àlgebra i la combinatòria hem demostrat un seguit de resultats que ens han permès caracteritzar un conjunt de solucions del problema. Finalment hem emprat aquesta caracterització per desenvolupar un algorisme eficient per resoldre el problema inicial plantejat en el domini de l'enginyeria. A més, hem aplicat l'algorisme a resoldre un altre problema de naturalesa enginyeril.

Bibliografia

Apostol, Tom M. (1998): *Introduction to Analytic Number Theory*. Springer.

Arazi, B. (1984): «Position recovery using binary sequences». *Electronics Letters*, volum 20: ps. 61–62.

Blyth, T. S.; Robertson, E. F. (2002*a*): *Basic Linear Algebra*. Springer.

— (2002*b*): *Further Linear Algebra*. Springer.

Bollman, Dorothy A. (1965): «Some periodicity properties of transformations on vectors spaces over residue class rings». *Journal Soc. Indust. Appl. Math.*, volum 13(3): ps. 902–912.

Bondy, J. A.; Murty, U. S. R. (1982): *Graph Theory with Applications*. Elsevier.

Cameron, Peter J. (1995): *Combinatorics: Topics, Techniques, Algorithms*. Cambridge University Press.

Fuertes, Josep M.; Balle, Borja; Ventura, Enric (2008): «Absolute type shaft encoding using LFSR sequences with prescribed length». *Apareixerà a IEEE Trans. Instrumentation and Measurement.*

von zur Gathen, Joachim; Gerhard, Jürgen (1999): *Modern computer algebra*. Cambridge University Press.

Golomb, Solomon W. (1981): *Shift Register Sequences*. Aegean Park Press.

Good, I. J. (1946): «Normal recurring decimals». *J. London Math.*, volum 21: ps. 167–172.

KHALFALLAH, H.; PETRIU, E. M.; GROEN, F. C. A. (1992): «Visual position recovery for an automated guided vehicle». *IEEE Trans. Instrumentation and Measurement Magazine*, volum 41(6): ps. 906–910.

LEMPEL, Abraham (1971): «m-ary closed sequences». *Journal of Combinatorial Theory*, volum 10: ps. 253–258.

LIDL, Rudolf; NIEDERREITER, Harald (1997): *Finite Fields*. Cambridge University Press.

MAYER, J. R. René (1999): «Optical encoder displacement sensors». Dins *The Measurement, Instrumentation, and Sensors Handbook*. CRC Press.

MCELIECE, Robert J. (1987): *Finite Fields for Computer Scientists and Engineers*. Kluwer Academic Publishers.

NIKU, Saeed B. (2001): *Introduction to Robotics. Analysis, Systems, Applications*. Prentice Hall.

PETRIU, E. M. (1985): «Abolute-type pseudorandom shaft encoder with any desired resolution». *Electronics Letters*, volum 21: ps. 215–216.

— (1998): «Absolute position measurement using pseudo-random binary encoding». *IEEE Instrumentation and Measurement Magazine*, volum 1(3): ps. 19–23.

POORTEN, Alf Van Der; *et al.* (2003): *Recurrence Sequences*. American Mathematical Society.

ROSENFELD, Vladimir R. (2002): «Enumerating de Bruijn sequences». *MATCH Commun. Math. Comput. Chem.*, volum 45: ps. 71–83.

TOMLINSON, G. H. (1987): «Abolute-type shaft encoder using shift register sequences». *Electronics Letters*, volum 23: ps. 398–400.

VENTURA, Enric (1997): «Dynamic structure of matrices over finite fields». Dins *Proceedings of EAMA-97*, ps. 413–420.