# 4

# Datu-moten espezifikazio ekuazionala

## 4.1 Datu-Mota Abstraktuak (DMA)

Programazio-lengoaia guztiek oinarrizko datu-motak erabiltzeko aukera ematen digute: osokoak, boolearrak, karaktereak ... Datu-mota horiek erabiliz egiten diren eragiketei buruz arrazoitzeko gai gara, bi arrazoi hauengatik:

- Eragiketa horien propietate algebraikoak ezagutzen ditugulako.
- Espresioak osatzean, erabil daitezkeen eragiketak eta beren portaera ondo definituta daudelako.

Adibidez, arrazoitzeko gaitasun hori dela medio, esan dezakegu hurrengo adierazpenak osokoak direla eta, gainera, baliokideak:

$$(x + y) + (z - 1)$$
$$(x + y + z) - 1$$

Zergatik? Osokoen gaineko batuketa elkarkorra delako. Badakigu, halaber, hurrengo bi adierazpen boolearrak baliokideak direla:

$$\neg\ (\ x\ \wedge\ y\ )$$
$$\neg x\ \vee\ \neg y$$

Are gehiago, aurreko propietateak datuen errepresentazioarekiko eta eragiketen inplementazioarekiko independenteak dira. Horregatik, errepresentazioa edo inplementazioa aldatu arren, gure arrazoibideek zuzenak izaten jarraituko dute.

Programazio-lengoaien oinarrizko datu-motez gain, programatzaileok askotan geure neurrira egindako datu-motak definitzen ditugu. Horrelakoetan ere, komeni da errepresentazioa eta inplementazio ondo bereiztea. Hori dela eta, *abstrakzioa* egiten dugula esango dugu; hau da, ez ditugula nahasten inplementazioari dagozkion xehetasunak portaeraren zehaztasunekin.

Abstrakzioarekin komenientzia horrek eraginda sortu zen *Datu-Mota Abstraktuen* (DMA) kontzeptua. Datu-mota abstraktuak erabiltzean, hortaz, lortzen dugu propietateak baliatzea inplementazioaren xehetasunei erreparatu gabe.

Laburbilduz, hauek dira DMAen erabileraren onura nagusiak:

- Betetzen diren propietateez arrazoitzeko aukera.
- DMA bat inplementatzen duten eta erabiltzen duten programen garapen independentea egin daiteke.

- DMA bat erabiltzen duten programak DMA horren inplementazio-aldaketekiko independenteak dira.

Onura horiei behar bezalako probetxua ateratzeko funtsezkoa da DMAen espezifikazio egokia egitea. Horrela lortuko da, eta ez bestela, DMA-a erabiltzen duten programatzaileen eta DMA-a inplementatzen dutenen artean komunikazio egokia eta lan-banaketa orekatua.

## 4.2 Espezifikazio ekuazionalaren oinarriak

DMAen espezifikazio-tekniken artean irakasgai honetarako aukeratu dugunari *espezifikazio ekuazionala* edo algebraikoa deritzo. Espezifikazio ekuazionalak bi atal biltzen ditu: *signatura* eta *ekuazioak*.

Batetik, *signaturak* motako adierazpenen (terminoen) sintaxia deskribatzen du. Horretarako, *motak* (nagusia eta lagungarriak) eta *eragiketak* deskribatzen dira. Eragiketa bakoitzeko

- izena
- zenbat eta zein motatako argumentuak
- emaitza

espezifikatu behar dira. Eragiketak bi multzo nagusitan bana daitezke:

- Eragiketa *eraikitzaileek* motako objektu guztiak definitzeko aukera ematen dute. Aldi berean, eraikitzaile *aratzak* (termino bakoitzak objektu desberdin bat adierazten du) edo *ez-aratzak* (objektu bat termino bat baino gehiagok adieraz dezakete) izan daitezke. Irakasgai honetan, eraikitzaile aratzak baino ez ditugu erabiliko.
- Eragiketa *ez-eraikitzaileak* objektuak definitzeko ez beste helburu batzuetarako erabiltzen dira (besteak beste, *objektuen gaineko kontsultak egiteko* edo *objektuetan aldaketak egiteko*).

Bestetik, *ekuazioek* signaturako eragiketen eragina deskribatzen dute, haien propietate algebraikoak deskribatuz. Orokorrean, hiru ekuazio mota erabiltzen dira:

- Berdinketak: $t_1 = t_2$
- Baldintzazkoak: $t_1 = t_2$ baldin $t_3 = t_4$
- Errore-ekuazioak: $t = errore$

Ekuazioak unibertsalki kuantifikatuta daudela suposatzen da. Hau da, ekuazio batean agertzen diren aldagai guztiak unibertsalak dira. Gainera, erroreak propagatzen dira: demagun $\frac{x}{0} = errorea$ ekuazioa dugula, orduan hurrengo ekuazio hau ere betetzen da:

$$(\ \frac{x}{0}\ )\ +\ y = errore$$

Adibidez, zenbaki arruntak errepresentatzen dituen *Nat* izeneko DMA-aren espezifikazio ekuazionala honako hau izan daiteke:

Mota: $Nat$
Lag: (ez dago)
Eragiketak:
  $zero$: $\to Nat$ (eraikitzailea)
  $hur$: $Nat \to Nat$ (eraikitzailea)
  $aurre$: $Nat \to Nat$ (aldaketa)
Ekuazioak:
  (1) $aurre(zero) = errore$ (errore-ekuazioa)
  (2) $aurre(hur(x)) = x$

DMA-aren objektuak honako hauek dira:

$$zero,\ hur(zero),\ hur(hur(zero)),\ \ldots$$

non denak desberdinak diren beraien artean. Erabilitako notazioak balio izan digu eragiketa eraikitzaileen izaera hobeto ulertzeko. Hala ere, zenbaki arrunten multzoarekin lan egiteko, *zero* eta *hur* eraikitzaileak erabiltzea baino erosoagoa zaigu ohiko zenbakien adierazpena erabiltzea. Paralelismo sinplea eginez, esan dezakegu *zero* eraikitzaile oinarrizkoa 0 balioa dela, eta *hur* eragiketa +1 eragiketaren baliokidea dela. Hau da, $zero$, $hur(zero)$, $hur(hur(zero))$, ... terminoak erabili ordez, $0$, $0+1$, $(0+1)+1$, ... terminoak erabil ditzakegu, edota, are eta sinpleago: $0$, $1$, $2$, ...

$Nat$ multzoaren espezifikazio sinplifikatua egin dugu, espezifikazio ekuazionalaren osagaiak zein diren eta nola erabiltzen diren erakusteko. Adibide batetik abiatuta, espezifikazio ekuazionalean erabiltzen diren teknika desberdin batzuk ikusiko ditugu orain.

## 4.3 Espezifikazio ekuazionalerako teknikak

Hasteko, aztertuko dugu nola erabili behar diren ekuazioak. Esan dugunez, motako objektuak zein diren adierazten dute eraikitzaileek. Eraikitzaileak erabiliz adieraz daitezkeen terminoak dira motako objektu *guztiak*. Hau da, ez dago motako objekturik eraikitzaileen bidez adierazi ezin daitekeenik. Eraikitzaileek definitzen dute objektuen unibertsioa. Horregatik, unibertsoa definitzeko ez da beharrezkoa ekuazioak erabiltzerik.

Eragiketa ez-eraikitzaileak, berriz, ekuazioen bidez definitzen dira, eta ekuazio horiek formulatzeko eraikitzaile bakoitzaren gaineko indukzioa erabiltzen da. Ohartu gaitezen propietate garrantzitsu batez: eraikitzaileek objektuen unibertso osoa adierazten dutenez, eragiketa ez-eraikitzaileek objektuen gainean zein portaera duten deskribatzeko, nahikoa da ekuazioen argumentu gisa eraikitzaileak erabiltzearekin. Azken batean, argumentuko eraikitzaile horiek motako objektu guztiak ordezkatzen dituzte. Hori dela eta, definitutako ekuazioak patroi moduan uler daitezke.

Gerta daiteke eragiketa ez-eraikitzaile bat definitu gabe egotea eraikitzaile baten gainean. Kasu horretan errore-ekuazio bat adierazi behar da.

Adibidez, $Nat$ izeneko DMA-aren espezifikazioan eragiketa ez-eraikitzaile bakar bat dago: *aurre* edo, baliokidea dena, $-1$. Eragiketa hori definitu gabe dago *zero* eraikitzailean (zenbaki negatibo bat aterako litzatekeelako, eta, dakigunez, zenbaki negatiboak ez dira $Nat$ multzokoak). Horregatik errore-ekuazio bat adierazten da. Baina, $+1$ eraikitzailea definituta dago, eta $-1$ eraikitzailearen portaera deskribatzeko berdinketa moduko ekuazio bat erabiltzen da.

Ekuazioen osaketarako oinarriak ikusi ditugula, gatozen *kontagailu* izeneko DMA-aren espezifikazio ekuazionala egitera. $Kontagailu$ motaren eta $Nat$ motaren artean analogia handiak daude. Espero dezagun, beraz, lehenengo espezifikazio ekuazional hau ulergarri gertatzea.

Lehenbiziko pausoan, signatura definituko dugu:

Mota: *kontagailu*
Lag: $Nat$
Eragiketak:
  *hutsa*: $\to$ *kontagailu*
  *gehitu*: *kontagailu* $\to$ *kontagailu*
  *gutxitu*: *kontagailu* $\to$ *kontagailu*
  *reset*: *kontagailu* $\to$ *kontagailu*
  *balioa*: *kontagailu* $\to$ *nat*

*hutsa* eta *gehitu* eragiketak eraikitzaileak dira, eta *gutxitu*, *reset* eta *balioa*, eragiketak ez-eraikitzaileak.

Ez-eraikitzaileak definitzeko, honako ekuazio hauek erabil daitezke:

Ekuazioak:
  (1) $gutxi(hutsa) = errore$ (errore-ekuazioa)
  (2) $gutxi(gehi(x)) = x$
  (3) $reset(hutsa) = hutsa$
  (4) $reset(gehi(x)) = hutsa$
  (5) $balioa(hutsa) = 0$
  (6) $balioa(gehi(x)) = balioa(x) + 1$

non *gutxitu* definitu gabe dagoen *hutsa* eraikitzailearen gainean, *reset* eragiketak beti kontagailu *hutsa* itzultzen duen, eta *balioa* eragiketak kontagailuaren balioa itzultzen duen $Nat$ DMA lagungarrian.

Osatu dugu *kontagailu* DMA-aren espezifikazio ekuazionala. DMA horren eragiketak ondo definituta daude, bai sintaktikoki (idazkerari dagokionez) bai semantikoki (duten egitekoari eta portaerari dagokionez).

Behin DMA bat definituta daukagula, beste eragiketa batzuk ere gehitu daitezke. Prozesu horrek *aberasketa* izena hartzen du. Eragiketa berriek aurretik definitutako DMAren bat hartzen dute oinarritzat eta beste DMA batzuk ere erabil ditzakete lagungarri gisa. Eragiketa ez-eraikitzaileak bezalaxe, eragiketa berriak ere oinarrizko DMA-aren eraikitzaileen gaineko ekuazioen bidez definitzen dira.

Adibidez, demagun *zeroa_da?* eragiketa gehitu nahi diogula lehen definitutako $Nat$ DMA-ari. Horretarako, *Bool* DMA-a ere erabiliko dugu:

Mota: $Nat$
Lag: *Bool*
Eragiketak:
  *zeroa_da?*: $Nat \to Bool$
Ekuazioak:
  (1) $zeroa\_da?(0) = true$
  (2) $zeroa\_da?(x + 1) = false$

Batzuetan ez da beharrezkoa beste DMA lagungarriren bat erabiltzea. Adibidez, eragiketa aritmetikoak gehitzeko:

Mota: $Nat$
Lag: (ez dago)
Eragiketak:
  $\_+\_$: $Nat \times Nat \to Nat$
Ekuazioak:
  (1) $0 + y = y$
  (2) $(x+1)+y = (x+y)+1$

Bukatzeko, DMA baten espezifikazioan egon litezke mota orokorrak, zehaztu gabekoak: *parametroak*. Datu-mota egituratuen eragiketak eta propietateak definitzeko, gehienetan ez da beharrezkoa elementuen mota ezagutzea. Horrelakoetan, elementuen mota aipatu nahi denean, *parametro* edo izen generiko bat erabiltzen da. Adibidez, hori gertatzen da array-en, pilen, ilaren, sekuentzien, arbolen eta abarren kasuan. Mota egituratu baten definizioan parametroak erabiltzeak esan nahi du egitura horren gaineko eragiketen definizioa independentea dela elementuen motarekiko.

Hurrengo sekzioan, parametro bat erabiltzen duten hiru DMA aztertuko ditugu: *sekuentzia*$(T)$, *pila*$(T)$ eta *arbit*$(T)$ (arbola bitarrak), non $T$ parametroak elementuen mota adierazten duen.

## 4.4 Datu-mota abstraktu oinarrizkoen espezifikazio ekuazionala

### 4.4.1 Sekuentziak

Atal honetan ikusiko dugun lehenengo DMA-a *sekuentzia*$(T)$ da. Sekuentziak elementuen segida linealak dira. Ez dute luzera finkorik edo aurre-definiturik. Sekuentziak osatzeko bi eraikitzaile erabiltzen dira: $\langle\,\rangle$ (sekuentzia hutsa) eta $\_\bullet\_$ (konposaketa). Adibidez, lehenengo bost zenbaki osokoen sekuentzia, eraikitzaileen bidez adierazita, honako hau da

$$1 \bullet (2 \bullet (3 \bullet (4 \bullet (5 \bullet \langle\,\rangle))))$$

baina normalean horrela errepresentatuko dugu:

$$\langle\ 1,\ 2,\ 3,\ 4,\ 5\ \rangle$$

Sekuentziekin erabiltzen diren eragiketa ohikoenak hauek dira: *lehena* (lehenengo elementu itzuli), *hondarra* (sekuentziaren lehenengo elementua kenduta geratzen den azpi-sekuentzia itzuli), *hutsa_da*?, $\in$ eta @ (kateaketa). Eragiketa hauek bilduko dituen *sekuentzia*$(T)$ DMA-aren espezifikazio ekuazionala honako hau da:

Mota: *sekuentzia*$(T)$
Lag: *Bool*
Eragiketak:
  $\langle\,\rangle$: $\to$ *sekuentzia*$(T)$ (eraikitzailea)
  $\_\bullet\_$: $T \times$ *sekuentzia*$(T) \to$ *sekuentzia*$(T)$ (eraikitzailea)
  *lehena*: *sekuentzia*$(T) \to T$ (aldaketa)
  *hondarra*: *sekuentzia*$(T) \to$ *sekuentzia*$(T)$ (aldaketa)
  *hutsa_da*?: *sekuentzia*$(T) \to$ *Bool* (kontsulta)
  $\_\in\_$: $T \times$ *sekuentzia*$(T) \to$ *Bool* (kontsulta)
  _@_: *sekuentzia*$(T) \times$ *sekuentzia*$(T) \to$ *sekuentzia*$(T)$ (aldaketa)

Ekuazioak:

(s.1) $lehena(\langle\ \rangle) = errorea$
(s.2) $lehena(x \bullet s) = x$
(s.3) $hondarra(\langle\ \rangle) = errorea$
(s.4) $hondarra(x \bullet s) = s$
(s.5) $hutsa\_da?(\langle\ \rangle) = true$
(s.6) $hutsa\_da?(x \bullet s) = false$
(s.7) $y \in \langle\ \rangle = false$
(s.8) $y \in (x \bullet s) = (x = y \vee y \in s)$
(s.9) $\langle\ \rangle @ s2 = s2$
(s.10) $(x \bullet s1)@s2 = x \bullet (s1@s2)$

Adibidez, izan bedi $1 \bullet (2 \bullet (3 \bullet \langle\ \rangle))$ sekuentzia, edo baliokidea den $\langle\ 1,\ 2,\ 3\ \rangle$ sekuentzia, espezifikazio ekuazionalean definitutako ekuazioak aplikatuz eta sekuentzia hori argumentu gisa harturik, horrela ebaluatuko lirateke honako espresio hauek:

$$lehena(\langle\ 1,\ 2,\ 3\ \rangle) = 1$$
$$hondarra(\langle\ 1,\ 2,\ 3\ \rangle) = \langle\ 2,\ 3\ \rangle$$
$$hutsa\_da?(\langle\ 1,\ 2,\ 3\ \rangle) = false$$
$$2 \in \langle\ 1,\ 2,\ 3\ \rangle = true$$
$$4 \in \langle\ 1,\ 2,\ 3\ \rangle = false$$
$$\langle\ 1,\ 2,\ 3\ \rangle @ \langle\ 4,\ 5\ \rangle = \langle\ 1,\ 2,\ 3,\ 4,\ 5\ \rangle$$

Gainera, beste eragiketa batzuk ere defini daitezke. Adibidez, $sekuentzia(T)$ $luzera$ eragiketarekin aberastu daiteke $Nat$ DMA lagungarria ere erabiliz:

Mota: $sekuentzia(T)$
Lag: $Nat$
Eragiketak:
$luzera$: $sekuentzia(T) \to Nat$
Ekuazioak:
(s.11) $luzera(\langle\ \rangle) = 0$
(s.12) $luzera(x \bullet s) = 1 + luzera(s)$

### 4.4.2 Pilak

$pila(T)$ DMA-a aztertuko dugu atal honetan. Pilak eta sekuentziak nahiko antzekoak dira: biak dira egitura linealak, luzera aldakorrekoak. Baina, portaerari erreparatuz gero, esan daiteke pilak mugatuagoak direla: piletan egituraren mutur bakarrean gertatzen dira kontsultak eta aldaketak. Mutur horri *gailur* deitzen zaio (bizitza errealeko pilen analogiaz). Piletan elementuak gailurretik sartu eta irteten dira, eta gailurreko elementua baizik ezin daiteke kontsultatu. Horrenbestez, sartutako azken elementua da ateratzen den lehena.

Adibidez, pilen errepresentazio grafiko ohikoa erabiliz (bertikala), honako hau da lortzen den pila lehenengo bost zenbaki arruntak sartutakoan, batetik hasiz eta bostean bukatuz:


Pilen datu-motaren eraikitzaileak *pilahutsa* eta *pilaratu* dira, eta beste eragiketa ohiko batzuk *gailurrekoa*, *despilatu* eta *hutsa_da*? dira. Formalki, hauxe da $pila(T)$ DMA-aren espezifikazio ekuazionala:

Mota: $pila(T)$
Lag: $Bool$
Eragiketak:
  $pilahutsa$: $\rightarrow pila(T)$
  $pilaratu$: $pila(T) \times T \rightarrow pila(T)$
  $gailurrekoa$: $pila(T) \rightarrow T$
  $despilatu$: $pila(T) \rightarrow pila(T)$
  $hutsa\_da?$: $pila(T) \rightarrow Bool$
Ekuazioak:
  (p.1) $gailurrekoa(pilahutsa) = errorea$
  (p.2) $gailurrekoa(pilaratu(p, x)) = x$
  (p.3) $despilatu(pilahutsa) = errorea$
  (p.4) $despilatu(pilaratu(p, x)) = p$
  (p.5) $hutsa\_da?(pilahutsa) = true$
  (p.6) $hutsa\_da?(pilaratu(p, x)) = false$

Pilekin lotutako beste eragiketa ohiko bat *altuera* da, pila bateko elementu kopurua itzultzen duena. Eragiketa honen espezifikazioa hauxe izan daiteke:

Mota: $pila(T)$
Lag: $Nat$
Eragiketak:
  $altuera$: $pila(T) \rightarrow Nat$
Ekuazioak:
  (p.7) $altuera(pilahutsa) = 0$
  (p.8) $altuera(pilaratu(p, x)) = 1 + altuera(p)$

### 4.4.3 Arbola bitarrak

Arbola bitarren DMA-a aztertuko dugu orain. Arbola bitarrak egitura hierarkikoa dira. Hona hemen arbola bitarren ezaugarri esanguratsuenak:

- Elementuak nodotan edo, arbolei dagokien terminologia baliatuz, adabegitan kokatzen dira.
- Adabegiak mailatan antolatzen dira, eta beren artean guraso-ume erlazioa definitzen da.
- *Erro* deritzo maila goreneko adabegiari, hau da, gurasorik ez duenari. Arbolek erro bakarra dute.
- Adabegiek ezin dezakete guraso bat baino gehiago eduki.

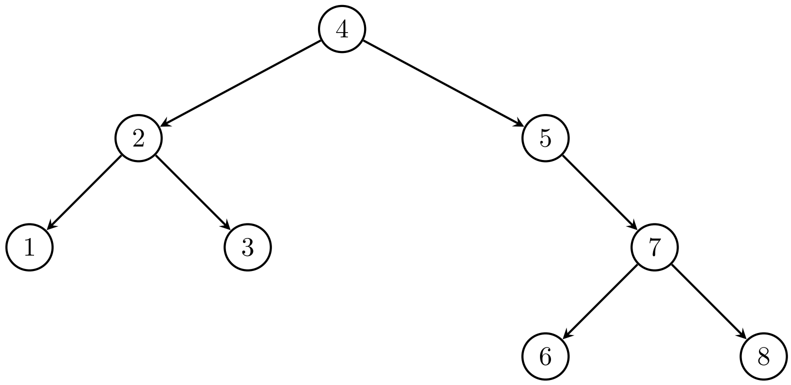
- Arbola bitarretako adabegiek, gehienez ere, bi ume izan ditzakete. Ume horiek ordenatuta daude, eta beren kokapenaren arabera, *ezkerreko* ala *eskuineko* ume deritze.

Ohikoa da arbola bitarrak errekurtsiboki definitzea. Hona definizio errekurtsiboa:

- Arbola hutsa (elementurik gabea), arbola bitarra da
- $e$ elementua eta $ezk$ eta $esk$ arbola bitarrak emanda, $ezk$ eta $esk$ arbolei $e$ elementua erro gisa jarriz (errotuz) lortzen dena ere arbola bitarra da.

Definizio errekurtsiboak oinarri egokia finkatzen du egituraren espezifikazio ekuazionalerako. Azken batean, ekuazioek molde induktiboa (errekurtsiboa) erabili ohi dutenez, datu-egitura molde berean definitzea lagungarria da espezifikaziorako.

Adibidez, irudiko arbola bitarrak 4 elementua dauka erroan, eta bi azpi-arbola ditu: ezkerreko azpi-arbolaren erroa 2 da, eta eskuinekoarena 5. Era berean, 2 erroko arbolak bere azpi-arbolak ditu, eta horrela definitzen da errekurtsiboki arbola osoa.




Gainera, arbola baten azpi-arbolak hutsak edo ez-hutsak izan daitezke. Biak hutsak direnean, arbolak *hosto* izena hartzen du, eta arbola bateko hosto-segida *muga* da. Irudiko arbolak lau hosto ditu: 1, 3, 6 eta 8 arbolak. Gerta daiteke, halaber, arbola baten azpi-arbola bat hutsa izatea eta bestea ez-hutsa: 5 adabegia erroan duen azpi-arbolari gertatzen zaio hori. Bukatzeko, errotik hosto batera joateko bidean dauden adabegien segidek *adarra* izena hartzen dute. Arbolaren sakontasuna 4koa da, adar luzeenaren luzera 4koa delako. Adibidez, hauek dira irudiko arbolaren adarrak:

$$\langle\, 4,\ 2,\ 1\,\rangle$$
$$\langle\, 4,\ 2,\ 3\,\rangle$$
$$\langle\, 4,\ 5,\ 7,\ 6\,\rangle$$
$$\langle\, 4,\ 5,\ 7,\ 8\,\rangle$$

Arbola bitarrak oso erabilgarriak dira, besteak beste, informazioaren bilaketan. Elementuen bilaketa azkarrak egiteko datu-mota egokia dira. Hain zuzen ere, *bilaketa dikotomikoa* egiteko aukera ematen dute. Bilaketa horiek egiteko, arbolak ere *dikotomikoa* izan behar du. Arbola dikotomikoetan ezkerreko azpi-arbolan agertzen diren elementu guztiak erroa baina txikiagoak dira, eta eskuineko azpi-arbolan agertzen direnak, handiagoak. Horrez gain, bilaketa dikotomikorako beharrezkoa da arbola *orekatuta* egotea. Arbola bitar bat orekatuta dagoela esango dugu baldin eta adabegi guztietarako betetzen bada ezkerreko azpi-arbolaren sakontasunaren eta eskuineko azpi-arbolaren sakontasunaren artean bateko aldea egotea gehienez. Propietate horiek betez gero, bilaketa-espazioa erdira mugatzen da bilaketa-pauso bakoitzean.

Behin arbola bitarren propietate eta erabilgarritasun nagusiak aipatuta, gatozen beren espezifikazio ekuazionala osatzera. Arbolen eraikitzaileak *hutsa* eta *errotu* dira, eta, horiekin batera, signaturan sartuko ditugu *erroa*, *ezker*, *eskuin* eta *hutsa_da*? eragiketak:

Mota: $arbit(T)$
Lag: $Bool$
Eragiketak:
  $hutsa$: $\to arbit(T)$
  $errotu$: $arbit(T) \times T \to arbit(T)$
  $erroa$: $arbit(T) \to T$
  $ezker$: $arbit(T) \to arbit(T)$
  $eskuin$: $arbit(T) \to arbit(T)$
  $hutsa\_da?$: $arbit(T) \to Bool$
Ekuazioak:
  (a.1) $erroa(hutsa) = errorea$
  (a.2) $erroa(errotu(x, ezk, esk)) = x$
  (a.3) $ezker(hutsa) = errorea$
  (a.4) $ezker(errotu(x, ezk, esk)) = ezk$
  (a.5) $eskuin(hutsa) = errorea$
  (a.6) $eskuin(errotu(x, ezk, esk)) = esk$
  (a.7) $hutsa\_da?(hutsa) = true$
  (a.8) $hutsa\_da?(errotu(x, ezk, esk)) = false$

Arbolen gaineko beste bi eragiketa ohikoak dira *sakon* (sakontasun maila) eta *adabegiKop* (adabegien kopurua). Arbola baten *sakona* adar luzeenaren luzera da, ikusi dugunez. Bi eragiketa hauek horrela espezifikatzen dira:

Mota: $arbit(T)$
Lag: $Nat$
Eragiketak:
  $sakon$: $arbit(T) \to Nat$
  $adabegiKop$: $arbit(T) \to Nat$
Ekuazioak:
  (a.9) $sakon(hutsa) = 0$
  (a.10) $sakon(errotu(x, ezk, esk)) = 1 + max(sakon(ezk), sakon(esk))$
  (a.11) $adabegiKop(hutsa) = 0$
  (a.12) $adabegiKop(errotu(x, ezk, esk)) = 1 + adabegiKop(ezk) + adabegiKop(esk)$

## 4.5 Propietateen frogapena

DMAen espezifikazio ekuazionalak aukera ematen du definitu diren eragiketen gaineko propietateak frogatzeko. Frogapen-metodo erabilgarriak izan daitezke bai *dedukzio bidezkoak* bai *indukzio bidezkoak*.

### 4.5.1 Dedukzio bidezko frogapenak

Dedukzio ekuazionalaren bidezko frogapenak sinplifikazioan oinarritzen dira.

Hona hemen sinplifikazio adibide bat:

$$
\begin{aligned}
& altuera(\underbrace{despilatu(pilaratu(pilaratu(pilahutsa, y), x))}_{(p.4)}) \\
&\quad = \underbrace{altuera(pilaratu(pilahutsa, y))}_{(p.8)} \\
&\quad = 1 + \underbrace{altuera(pilahutsa)}_{(p.7)} \\
&\quad = 1 + 0 \\
&\quad = 1
\end{aligned}
$$

Sinplifikazioak erabiliz, gatozen frogatzera honako propietate hau:

Edozein $P$ pilatarako, $P$ hutsa ez bada, orduan

$$P = pilaratu(despilatu(P), gailurrekoa(P))$$

Propietatearen frogapena: $P$ hutsa ez denez, orduan $P = pilaratu(K, x)$, non $K$ pila bat den ($K \in pila(T)$) eta $x$ edozein elementu ($x \in T$). Beraz:

$$
\begin{aligned}
& pilaratu(despilatu(P), gailurrekoa(P)) \\
&\quad = pilaratu(despilatu(pilaratu(K, x)), \underbrace{gailurrekoa(pilaratu(K, x))}_{(p.2)}) \\
&\quad = pilaratu(\underbrace{despilatu(pilaratu(K, x))}_{(p.4)}, x) \\
&\quad = pilaratu(K, x) \\
&\quad = P
\end{aligned}
$$

### 4.5.2 Indukzio bidezko frogapenak

Indukzio bidezko frogapenak DMA-aren eraikitzaileen gainean egiten dira.

Esate baterako, edozein $S_1$ eta $S_2$ sekuentziatarako, honako propietate hau froga daiteke:

$$luzera(S_1 @ S_2) = luzera(S_1) + luzera(S_2)$$

Hasteko, erabaki behar da eragiketaren zein argumenturen gainean erabiliko den indukzioa. Kasu honetan, $S_1$ sekuentziaren gainean erabiliko dugu.

<u>Oinarrizko kasua</u> $S_1 = \langle\ \rangle$ da:

$$
\begin{aligned}
& luzera(S_1 @ S_2) \\
&\quad = luzera(\underbrace{\langle\ \rangle @ S_2}_{(s.9)}) \\
&\quad = luzera(S_2) \\
&\quad = \underbrace{0}_{s.11} + luzera(S_2) \\
&\quad = luzera(\langle\ \rangle) + luzera(S_2) \\
&\quad = luzera(S_1) + luzera(S_2)
\end{aligned}
$$

Indukzio-kasua $S_1 = x \bullet S$ hartuko dugu, non $x \in T$ eta $S \in sekuentzia(T)$. Indukzio-hipotesia honako hau da:

$$(\mathbf{I.H.}) \qquad luzera(S@S_2) = luzera(S) + luzera(S_2)$$

Indukzio-kasuaren frogapena:

$$\begin{aligned}
&luzera(S_1@S_2)\\
&\quad= luzera(\underbrace{(x \bullet S)@S_2}_{(s.10)})\\
&\quad= \underbrace{luzera(x \bullet (S@S_2))}_{(s.12)}\\
&\quad= 1 + \underbrace{luzera(S@S_2)}_{(I.H.)}\\
&\quad= 1 + (luzera(S) + luzera(S_2))\\
&\quad= \underbrace{(1 + luzera(S))}_{(s.12)} + luzera(S_2)\\
&\quad= luzera(x \bullet S) + luzera(S_2)\\
&\quad= luzera(S_1) + luzera(S_2)
\end{aligned}$$