

## **6. Gaia. Sistema Adimendunak**

### Edukia

6.1. Sarrera.....	1
6.2. Hiztegia.....	2
6.3. DNF monotonoak .....	3
6.3.1. Egoera.....	3
6.3.2. Adibideak .....	4
6.3.3. Algoritmoak.....	10
6.3.4. Algoritmoek erabiltzaileari egin diezazkiokeen galdera motak .....	10
6.4. k-CNFak .....	11
6.4.1. Egoera.....	11
6.4.2. Adibidea .....	12
6.4.3. Algoritmoa.....	13
6.4.4. Algoritmoak erabiltzaileari egin diezazkiokeen galdera motak .....	13
6.5. k-DNFak .....	14
6.5.1. Egoera.....	14
6.5.2. Adibidea .....	15
6.5.3. Algoritmoa.....	16
6.5.4. Algoritmoak erabiltzaileari egin diezazkiokeen galdera motak .....	16

### **6.1. Sarrera**

Gai honetan jokaera adimenduna erakusten edo simulatzen duten hiru algoritmo ikusiko ditugu. Sistema adimendunak, azken batean, programak dira eta erabiltzaileari sistema hauen jokaerak adimenduna dela iruditzen baldin bazaio ere, programatzaileak jokaera hori programatu behar izan du.

Erabiltzaileari pistak eskatu eta logika proposizionaleko hiru formula-mota asmatzeko gai diren hiru algoritmo aztertuko ditugu.

Gai honetan ikusiko diren hiru algoritmoak polinomikoak dira eta ondorioz nahiko onak konplexutasunari eta eranginkortasunari dagokionez. Logika proposizionaleko beste formula-mota batzuentzat oraindik ez da aurkitu mota horietako formulak asmatzeko gai den algoritmo polinomikorik. Honelako algoritmo esponentzialak diseinatzea erraza da baina algoritmo esponentzialak ez dira erabilgarriak. Izan ere, formula bat asmatzeko ehundaka edo milaka urte behar dituen programa bat edukitzea alferrikakoa baita oro har. Beraz, arlo honetan ikerketa irekita dago, algoritmo polinomikoak aurkitzearen arazoa hor baitago formula-mota askorentzat.

Gai honetan formula logikoak asmatzeko ikusiko ditugun algoritmoen antzekoak automata finituak asmatzeko eta beste kontzeptu matematiko batzuk asmatzeko ere badaude. Baina kontzeptu askorentzat oraindik ez da algoritmo polinomikorik aurkitu.

## 6.2. Hiztegia

Orain, algoritmoen azalpenean erabiliko den hiztegia laburtuko da:

- **Literala:** aldagai bat edo ukatutako aldagai bat

1. adibidea:  $x_4$
2. adibidea:  $\neg x_4$

- **Terminoak:** literalen konjuntzioa (eta logikoa)

Adibidea:  $(\neg x_1 \wedge x_2 \wedge \neg x_5)$

- **Klausula:** literalen disjuntzioa (edo logikoa)

Adibidea:  $(x_1 \vee \neg x_2 \vee x_4)$

- **k-CNF-a:** gehienez  $k$  literal izan ditzaketen klausulez osatutako konjuntzioa.
- **k-DNF-a:** gehienez  $k$  literal izan ditzaketen terminoz osatutako disjuntzioa.
- **Balorazioa:** balorazio bat aldagai boolear bakoitzari True edo False balioa esleitzen dion espresio bat da.

Adibidea:

$\varphi = (x_1 \vee x_2) \wedge (x_2 \vee x_3)$  formula emanda,  $v_1 = (\text{True}, \text{True}, \text{True})$  eta  $v_2 = (\text{True}, \text{False}, \text{False})$  bi balorazio dira. Hiru aldagai desberdin ditugunez, balorazioek hiru osagai izango dituzte, aldagai bakoitzari balioa emateko  $(x_1, x_2, x_3)$  ordenean. Adibide honetan  $v_1$  balorazioarentzat  $\varphi$  formularen balioa True da eta  $v_2$  balorazioarentzat  $\varphi$  formularen balioa False da. Hiru aldagai izanda, guztira 8 konbinazio sor daitezke:  $2^3$

- **Formula baliokideak:**  $\varphi$  eta  $\psi$  bi formula emanda,  $\varphi$  eta  $\psi$  baliokideak direla esaten da balorazio guztientzat  $\varphi$  formularen balioa eta  $\psi$  formularen balioa berdinak badira.

Adibidea:

- $\neg(x_1 \wedge x_2 \wedge x_3)$  eta  $(\neg x_1 \vee \neg x_2 \vee \neg x_3)$  formulak baliokideak dira.
- $(x_1 \vee x_2) \wedge x_2 \wedge x_3$  eta  $x_2 \wedge x_3$  formulak baliokideak dira.
- $(x_1 \vee x_2) \wedge (x_2 \vee x_3)$  eta  $(x_1 \vee x_2) \wedge x_3$  ez dira baliokideak  $v = (\text{False}, \text{True}, \text{False})$  baloraziorako  $(x_1 \vee x_2) \wedge (x_2 \vee x_3)$  formula True baita eta  $(x_1 \vee x_2) \wedge x_3$  formula aldiz False.

### **6.3. DNF monotonoak**

Formula boolear bat DNF monotonoa dela esaten da negaziorik ez duten aldagaien konjuntzioak diren azpiformulen disjuntzio bat baldin bada.

Adibidea: honako formula hau DNF monotonoa da

$$(x_3) \vee (x_1 \wedge x_2 \wedge x_4) \vee (x_2 \wedge x_5) \vee (x_1 \wedge x_5)$$

#### **6.3.1. Egoera**

- Erabiltzaileak DNF monotono bat du buruan (formula horri  $g$  deituko diogu). Formula horretan  $x_1, x_2, \dots, x_n$  aldagaiak ager daitezke. Hala ere garbi eduki behar da gerta daitekeela  $x_1, x_2, \dots, x_n$  tarteko aldagairen batzuk ez agertzea  $g$  formularen. Bestalde  $1 \leq n$  beteko da beti.

Esate baterako,  $n$ -ren balioa 6 denean erabiltzaileak honako formula hau eduki dezake buruan:

$$g = (x_3) \vee (x_2 \wedge x_4) \vee (x_1 \wedge x_4 \wedge x_6)$$

Formula horretan  $x_5$  ez da agertzen.

- Hasteko, algoritmoak (Sistema adimendunak)  **$n$ -ren balioa eskatuko** dio erabiltzaileari.
- Gero, algoritmoa asmatze-prozesuarekin hasiko da:
  - Algoritmoak **hasierako dnf-a sortuko du**. Algoritmoak sortutako hasierako dnf horri  **$h$**  deituko diogu eta formula horren balioa False izango da (False disjuntzio hutsa da).
  - Algoritmoak  $h$  formula erabiltzaileari aurkeztuko dio. Erabiltzaileak  $g$  eta  $h$  baliokideak al diren erantzun behar du (Bai edo Ez erantzun behar du).
  - $g$  eta  $h$  formulak baliokideak ez badira,  $g$  True eta  $h$  False egiten duen  $\vee$  balorazio bat eskatuko dio algoritmoak erabiltzaileari. (Kontuz!!  $k$ -dnf-etan hau alderantziz da)
  - Balorazio hori kontuan hartuz algoritmoak  $h$  berri bat eraikiko du.
  - Prozesua algoritmoak  $g$ -ren baliokidea den  $h$  formula bat eraiki arte errepikatuko da.

### 6.3.2. Adibideak

- 1. Adibidea:

- Erabiltzaileak  $g = (x_2 \wedge x_3) \vee (x_1)$  formula du buruan
- $n = 3$  da
- Algoritmoak asmatze-prozesuari ekingo dio eta honako formula hau proposatuko du:

$$h = \text{False}$$

- $g$  eta  $h$  baliokideak ez direnez, erabiltzaileak  $g$  True eta  $h$  False egiten duen  $v$  balorazio bat eman behar du:

Demagun erabiltzaileak  $v = (\text{True}, \text{False}, \text{True})$  balorazioa ematen duela.  $(\text{True}, \text{False}, \text{True})$  hirukoteak  $(x_1, x_2, x_3)$  aldagaientzat balioak zein diren zehazten du.

- Jarraian algoritmoak  $g$ -ri dagokion  $t$  **inplikatzaile lehena** kalkulatu behar du:

- ✓ Badakigu  $v$  balorazioarentzat  $g$  formula True dela. Orain True gutxiago dituen eta  $g$  True egiten duen balorazioen bat ba al dagoen jakin nahi da. Horretarako  $v$  balorazioko True balioak banan-banan False balioaz ordezkatzuz frogak egin beharko dira, True gutxiago dituen eta hala ere  $g$  True egiten duen balorazio bat aurkitu arte edo True gutxiago dituzten balorazio denek  $g$  False egiten dutela ziur egon arte. Helburua ahalik eta True gehien False balioaz ordezkatzea da. Frogen prozesu hau honela egingo da:
- ✓ Algoritmoak  $v$ -ko lehenengo True balioa False balioaz ordezkatzuko du:  $v = (\text{False}, \text{False}, \text{True})$ .
- ✓ Orain  $v$  balorazioak  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
- ✓ Erabiltzaileak ezetz erantzungo du eta ondorioz, aldaketa desegin behar da eta berriro  $v = (\text{True}, \text{False}, \text{True})$  edukiko dugu.
- ✓ Jarraian algoritmoak hurrengo True balioa False balioaz ordezkatzuko du. Beraz, hirugarren posizioan dagoen True balioa False balioaz ordezkatzuko du:  $v = (\text{True}, \text{False}, \text{False})$ .
- ✓ Orain  $v$  balorazio berri honek  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
- ✓ Erabiltzaileak baietz erantzungo du eta ondorioz  $v$  balorazioan egindako aldaketa hau mantendu egingo da eta  $v$  honela geldituko da:  $v = (\text{True}, \text{False}, \text{False})$ .
- ✓  $v$  balorazioan True balioa zuten posizio denak aztertu ditugunez, froga hauekin bukatu da eta orain  $t$  inplikatzaile lehena eraikiko da  $v = (\text{True}, \text{False}, \text{False})$  balorazioan True balioa duten aldagaien konjuntzio gisa:  $t = x_1$ .

- Algoritmoak h formula berri bat proposatuko du soluziotzat lehengo h formulari t formula disjuntzio baten bidez erantsiz:

$$h = \text{False} \vee t \quad \text{hau da} \quad h = \text{False} \vee x1$$

- g eta h baliokideak al diren galdetuko dio algoritmoak erabiltzaileari. Baliokideak ez direnez, g formula True eta h formula False egiten duen v balorazio bat emango du erabiltzaileak:

$$v = (\text{False}, \text{True}, \text{True})$$

- Jarraian algoritmoak g formularen **beste t inplikatzailerik lehen bat** kalkulatu behar du v balorazio berrian oinarrituz:

- ✓ Badakigu v balorazioarentzat g formula True dela. Orain True gutxiago dituen eta g True egiten duen balorazioen bat ba al dagoen jakin nahi da. Horretarako, v balorazioko True balioak banan-banan False balioaz ordezkatzuz frogak egin beharko dira True gutxiago dituen eta hala ere g True egiten duen balorazio bat aurkitu arte edo True gutxiago dituzten balorazio denek g False egiten dutela ziur egon arte. Helburua ahalik eta True gehien False balioaz ordezkatzea da. Frogen prozesu hau honela egingo da.
- ✓ Algoritmoak lehenengo True balioa False balioaz ordezkatzeko du v balorazioan:  $v = (\text{False}, \mathbf{\text{False}}, \text{True})$ .
- ✓ Gero, v balorazio honek g formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
- ✓ Erabiltzaileak ezetz erantzungo du eta, ondorioz, aldaketa desegin behar da eta v honela geldituko da:  $v = (\text{False}, \mathbf{\text{True}}, \text{True})$ .
- ✓ Algoritmoak  $v = (\text{False}, \text{True}, \text{True})$  balorazioa hartu eta beste True bat False balioaz trukatzeko saiaturik da. Oraingoan hirugarren posizioan dagoen True balioa ordezkatzeko du False balioaz:  $v = (\text{False}, \text{True}, \mathbf{\text{False}})$ .
- ✓ Jarraian, v balorazio honek g formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
- ✓ Erabiltzaileak ezetz erantzungo du eta ondorioz aldaketa desegin behar da eta v honela geldituko da:  $v = (\text{False}, \mathbf{\text{True}}, \text{True})$ .
- ✓ v balorazioan True balioa zuten posizio denak aztertu ditugunez, froga hauekin bukatu da eta orain t inplikatzailerik lehenena eraikiko da  $v = (\text{False}, \text{True}, \text{True})$  balorazioan True balioa duten aldagaien konjuntzio gisa:  $t = x2 \wedge x3$ .

- t hori disjuntzio baten bidez lehengo h-ri erantsiz h berri bat proposatuko du algoritmoak:

$$h = \text{False} \vee x1 \vee (x2 \wedge x3)$$

- g eta h baliokideak al diren galdetuko dio algoritmoak erabiltzaileari.
- h eta g baliokideak direnez, asmatze-prozesua bukatu da. Algoritmoak g formula asmatu du.

**OHARRA:** Teknikoki  $h$  formulako False hori ken daiteke. Izan ere, False disjuntzio hutsa da eta lehenengo inplikatzailer lehenaren eranstean, disjuntzio hutsa edukizetik osagai bakarra duen disjuntzio bat edukitzera pasatzen gara.

$$\text{False} \rightarrow x_1 \rightarrow x_1 \vee (x_2 \wedge x_3)$$

• **2. adibidea:**

- Erabiltzaileak  $g = (x_1 \wedge x_2) \vee (x_3) \vee (x_1 \wedge x_5)$  formula du buruan.
- $n = 5$  da.
- Asmatze-prozesuarekin hasteko algoritmoak honako formula hau proposatzen du:

$$h = \text{False}$$

- Jarraian, erabiltzaileari  $h$  eta  $g$  baliokideak al diren galdetuko dio algoritmoak. Erabiltzaileak ezetz erantzungo dio eta  $g$  True eta  $h$  False egiten duen  $v$  balorazio bat emango du. Demagun erabiltzaileak honako balorazio hau eman duela:

$$v = (\text{True}, \text{True}, \text{False}, \text{True}, \text{True})$$

- Orain algoritmoak  $t$  **inplikatzailer lehen bat kalkulatu** behar du. Inplikatzailer lehen hori honako urratsei jarraituz kalkulatu da:

- ✓ Badakigu  $v$  balorazioarentzat  $g$  formula True dela. Orain True gutxiago dituen eta  $g$  True egiten duen balorazioen bat ba al dagoen jakin nahi da. Horretarako,  $v$  balorazioko True balioak, banan-banan, False balioaz ordezkatzuz frogak egin beharko dira True gutxiago dituen eta, hala ere,  $g$  True egiten duen balorazio bat aurkitu arte edo True gutxiago dituzten balorazio denek  $g$  False egiten dutela ziur egon arte. Helburua ahalik eta True gehien False balioaz ordezkatzea da. Frogen prozesu hau honela burutuko da.
- ✓ Algoritmoak lehenengo True balioa False balioaz ordezkatu du eta  $v$  honela gelditu da:  $v = (\text{False}, \text{True}, \text{False}, \text{True}, \text{True})$ .
- ✓ Jarraian,  $v$  balorazio honek  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
- ✓ Erabiltzaileak ezetz erantzungo du eta ondorioz aldaketa desegin behar da eta  $v$  honela gelditu da:
 
$$v = (\text{True}, \text{True}, \text{False}, \text{True}, \text{True}).$$
- ✓  $v$  balorazioko hurrengo True balioarekin frogatuko du algoritmoak:  $v = (\text{True}, \text{False}, \text{False}, \text{True}, \text{True})$ .
- ✓ Jarraian  $v$  balorazio honek  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
- ✓ Erabiltzaileak baietz erantzungo du eta ondorioz  $v$  balorazioan egindako aldaketa hau mantendu egingo da eta  $v$  momentuz honela gelditu da:  $v = (\text{True}, \text{False}, \text{False}, \text{True}, \text{True})$ .

- ✓ Orain hurrengo True balioarekin frogatuko da:  
 $v = (\text{True}, \text{False}, \text{False}, \text{True}, \text{True})$ .
  - ✓ Algoritmoak laugarren posizoko True balioa False balioaz ordezkatzuko du:  $v = (\text{True}, \text{False}, \text{False}, \text{False}, \text{True})$ .
  - ✓ Jarraian  $v$  balorazio honek  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
  - ✓ Erabiltzaileak baietz erantzungo du eta, ondorioz,  $v$  balorazioan egindako aldaketa hau mantendu egingo da eta  $v$  momentuz honela geldituko da:  $v = (\text{True}, \text{False}, \text{False}, \text{False}, \text{True})$ .
  - ✓ Hurrengo True balioaren txanda da orain:  $v = (\text{True}, \text{False}, \text{False}, \text{False}, \text{True})$ .
  - ✓ Bosgarren posizioako True balioa False balioaz ordezkatzuko du algoritmoak:  $v = (\text{True}, \text{False}, \text{False}, \text{False}, \text{False})$ .
  - ✓ Jarraian,  $v$  balorazio honek  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
  - ✓ Erabiltzaileak ezetz erantzungo du eta, ondorioz, azkeneko aldaketa hau desegin behar da eta  $v$  honela geldituko da:  $v = (\text{True}, \text{False}, \text{False}, \text{False}, \text{True})$ .
  - ✓ Algoritmoak  $v$  balorazioako True denak aztertu dituenaz, frogekin bukatu da eta  $t$  inplikatzailerik lehena True balioa duten  $v$  balorazioako aldagaien konjuntzioa izango da:  $t = (x1 \wedge x5)$
- Algoritmoak  $t$  inplikante lehena  $h$  formulari erantsiko dio eta  $h$  berria proposatzuko du:
- $$h = \text{False} \vee (x1 \wedge x5)$$
- $g$  eta  $h$  baliokideak al diren galdetuko dio algoritmoak erabiltzaileari. Baliokideak ez direnez,  $g$  formula True eta  $h$  formula False egiten duen  $v$  balorazio bat emango du erabiltzaileak:
- $$v = (\text{True}, \text{True}, \text{False}, \text{True}, \text{False})$$
- Algoritmoak  $g$  formularen **beste inplikatzailerik lehen bat kalkulatu** behar du orain  $v$  berri honetan oinarrituz.  $t$  inplikatzailerik lehen berri hori honako urratsei jarraituz kalkulatzeko da:
- ✓ Badakigu  $v$  balorazioarentzat  $g$  formula True dela. Orain True gutxiago dituen eta  $g$  True egiten duen balorazioaren bat ba al dagoen jakin nahi da. Horretarako,  $v$  balorazioako True balioak, banan-banan, False balioaz ordezkatzuz frogak egin beharko dira True gutxiago dituen eta, hala ere,  $g$  True egiten duen balorazio bat aurkitu arte edo True gutxiago dituzten balorazio denek  $g$  False egiten dutela ziur egon arte. Helburua, ahalik eta True gehien False balioaz ordezkatzea da. Frogen prozesu hau honela egingo da.
  - ✓ Algoritmoak  $v$  balorazioako lehenengo True balioa False balioaz ordezkatzuko du:  $v = (\text{False}, \text{True}, \text{False}, \text{True}, \text{False})$ .
  - ✓ Jarraian,  $v$  balorazio honek  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
  - ✓ Erabiltzaileak ezetz erantzungo du eta, ondorioz, aldaketa hau desegin behar da eta  $v$  honela geldituko da:

$v = (\mathbf{True}, \text{True}, \text{False}, \text{True}, \text{False}).$

- ✓ Bigarren posizioako True balioa False balioaz ordezkatu du algoritmoak eta  $v$  honela geldituko da:

$v = (\text{True}, \mathbf{False}, \text{False}, \text{True}, \text{False}).$

- ✓ Jarraian,  $v$  balorazio honek  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
- ✓ Erabiltzaileak ezetz erantzungo du eta, ondorioz, aldaketa hau desegin behar da eta  $v$  honela geldituko da:

$v = (\text{True}, \mathbf{True}, \text{False}, \text{True}, \text{False}).$

- ✓ Jarraian, laugarren posizioako True balioa False balioaz ordezkatu du algoritmoak eta  $v$  honela geldituko da:  $v = (\text{True}, \text{True}, \text{False}, \mathbf{False}, \text{False}).$
- ✓ Jarraian,  $v$  balorazio honek  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
- ✓ Erabiltzaileak baietz erantzungo du eta ondorioz  $v$  balorazioan egindako aldaketa hau mantendu egingo da eta  $v$  momentuz honela geldituko da:  $v = (\text{True}, \text{True}, \text{False}, \mathbf{False}, \text{False}).$
- ✓ Algoritmoak  $v$  balorazioako True denak aztertu dituenez, frogekin bukatu da eta  $t$  inplikatzailerik lehenen True balioa duten  $v$  balorazioak aldagaien konjuntzioa izango da:  $t = (x_1 \wedge x_2).$

- Algoritmoak  $t$  inplikatzailerik lehenen  $h$  formulari erantsiko dio eta  $h$  berria proposatuko du:

$h = \text{False} \vee (x_1 \wedge x_5) \vee (x_1 \wedge x_2)$

- $g$  eta  $h$  baliokideak al diren galdetuko dio algoritmoak erabiltzaileari. Baliokideak ez direnez,  $g$  formula True eta  $h$  formula False egiten duen  $v$  balorazio bat emango du erabiltzaileak:

$v = (\text{True}, \text{False}, \text{True}, \text{True}, \text{False})$

- Algoritmoak  $g$  formularen **beste inplikatzailerik lehen bat kalkulatu** behar du orain  $v$  berri honetan oinarrituz.  $t$  inplikante berri hori honako urratsei jarraituz kalkulatu da:

- ✓ Badakigu  $v$  balorazioarentzat  $g$  formula True dela. Orain True gutxiago dituen eta  $g$  True egiten duen balorazioen bat ba al dagoen jakin nahi da. Horretarako  $v$  balorazioako True balioak, banan-banan, False balioaz ordezkatzuz frogak egin beharko dira True gutxiago dituen eta, hala ere,  $g$  True egiten duen balorazio bat aurkitu arte edo True gutxiago dituzten balorazio denek  $g$  False egiten dutela ziur egon arte. Helburua ahalik eta True gehien False balioaz ordezkatzea da. Frogen prozesu hau honela egingo da.
- ✓ Algoritmoak  $v$  balorazioako lehenengo True balioa False balioaz ordezkatu du:  $v = (\mathbf{False}, \text{False}, \text{True}, \text{True}, \text{False}).$
- ✓ Jarraian,  $v$  balorazio honek  $g$  formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
- ✓ Erabiltzaileak baietz erantzungo du eta, ondorioz,  $v$  balorazioan egindako aldaketa hau mantendu egingo da eta  $v$ , momentuz, honela geldituko da:  $v = (\text{False}, \text{False}, \text{True}, \text{True}, \text{False}).$



- ✓ Jarraian, algoritmoak v-ko hurrengo True balioa False balioaz ordezkatu du:  $v = (\text{False}, \text{False}, \mathbf{\text{False}}, \text{True}, \text{False})$ .
  - ✓ Jarraian, v balorazio honek g formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
  - ✓ Erabiltzaileak ezetz erantzungo du eta ondorioz aldaketa hau desegin behar da eta v honela geldituko da:  $v = (\text{False}, \text{False}, \mathbf{\text{True}}, \text{True}, \text{False})$ .
  - ✓ Orain algoritmoak hurrengo True balioa ordezkatu du:  $v = (\text{False}, \text{False}, \text{True}, \mathbf{\text{False}}, \text{False})$ .
  - ✓ Jarraian, v balorazio honek g formula True egiten al duen galdetuko dio algoritmoak erabiltzaileari.
  - ✓ Erabiltzaileak baietz erantzungo du eta ondorioz v balorazioan egindako aldaketa hau mantendu egingo da eta v, momentuz, honela geldituko da:  $v = (\text{False}, \text{False}, \text{True}, \text{False}, \text{False})$ .
  - ✓ Algoritmoak v balorazioko True denak aztertu dituenaz, frogekin bukatu da eta t inplikatzailerik lehena True balioa duten  $v = (\text{False}, \text{False}, \text{True}, \text{False}, \text{False})$  balorazioko aldagaien konjuntzioa izango da:  $t = x3$ .
- Algoritmoak t inplikatzailerik lehena h formulari erantsiko dio eta h berria proposatu du:
- $$h = \text{False} \vee (x1 \wedge x5) \vee (x1 \wedge x2) \vee (x3)$$
- g eta h baliokideak al diren galdetuko dio algoritmoak erabiltzaileari.
- h eta g baliokideak direnez, asmatze-prozesua bukatu da. Algoritmoak g formula asmatu du:  $h = \text{False} \vee (x1 \wedge x5) \vee (x1 \wedge x2) \vee (x3)$

**OHARRA:** Teknikoki h formulako False hori ken daiteke. Izan ere, False disjuntzio hutsa da eta lehenengo inplikatzailerik lehena eranstea, disjuntzio hutsa edukitzetik osagai bakarra duen disjuntzio bat edukitzera pasatzen gara.

$$\text{False} \rightarrow (x1 \wedge x5) \rightarrow (x1 \wedge x5) \vee (x1 \wedge x2) \rightarrow (x1 \wedge x5) \vee (x1 \wedge x2) \vee (x3)$$

**6.3.3. Algoritmoak**

- **DNF monotonoen algoritmoa:**

$\{n \geq 1 \wedge \text{dnf\_monotonoa\_da}(g)\}$

$h := \text{False};$

$\text{baliokid} := \text{erabiltzaileari\_formula\_baliokidea\_al\_den\_galdetu}(h);$

**while not baliokid loop**

$v := \text{erabiltzaileari\_balorazioa\_eskatu}();$

$t := \text{inplikatzaile\_lehena\_kalkulatu}(v);$

$h := \text{inplikatzaile\_lehena\_erantsi}(h, t);$

$\text{baliokid} := \text{erabiltzaileari\_formula\_baliokidea\_al\_den\_galdetu}(h);$

**end loop;**

$\{\text{dnf\_monotonoa\_da}(h) \wedge h \leftrightarrow g\}$

- **"inplikatzaile\_lehena\_kalkulatu" izeneko funtzioari dagokion algoritmoa:**

$\{n \geq 1 \wedge \text{balorazioa\_da}(v)\}$

$i := 1;$

**while i ≤ n loop**

**if**  $v(i) = \text{True}$  **then**

$v(i) := \text{False};$

$r := \text{formula\_egiazkoa\_al\_den\_galdetu\_erabiltzaileari}(v);$

**if**  $r = \text{False}$  **then**  $v(i) := \text{True};$  **end if;**

**end if;**

$i := i + 1;$

**end loop;**

$t := \text{inplikatzaile\_lehena\_eraiki}(v);$  *{v balorazioan True balioa duten aldagaien konjuntzioa }*

$\{\text{inplikatzaile\_lehena\_da}(t, g)\}$

**6.3.4. Algoritmoek erabiltzaileari egin diezazkiokeen galdera motak**

Algoritmo honek hiru eratako galderak egiten dizkio erabiltzaileari:

- ✓  $g$  True eta  $h$  False egiten dituen  $v$  balorazio bat eskatu.
- ✓  $h$  eta  $g$  baliokideak al diren galdetu
- ✓ Inplikatzaile lehen bat aurkitzeko prozesuan  $v$  aldatzen denean,  $v$  berriak  $g$  formula True egiten al duen galdetu.

## 6.4. k-CNFak

Formula boolear bat k-CNFa dela esaten da literalen disjuntzioak diren azpiformulen konjuntzio bat baldin bada eta disjuntzio bakoitzak gehienez k literal baldin baditu. Literal bat aldagai bat edo ukatutako aldagai bat da.

Honako formula hau 3-CNFa da:

$$(\neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_4)$$

### 6.4.1. Egoera

- Erabiltzaileak k-CNF bat du buruan (formula horri g deituko diogu). Formula horretan  $x_1, x_2, \dots, x_n$  aldagaiak ager daitezke. Hala ere, garbi eduki behar da gerta daitekeela  $x_1, x_2, \dots, x_n$  tarteko aldagairen batzuk ez agertzea g formularen. Bestalde  $1 \leq k \leq n$  beteko da beti.

Esate baterako  $k = 2$  baldin bada eta  $n = 4$  baldin bada, erabiltzaileak honako formula hau izan dezake buruan:

$$g = (\neg x_3) \wedge (x_2 \vee x_3) \wedge (x_4)$$

Formula horretan  $x_1$  ez da agertzen.

- Hasteko, algoritmoak (Sistema adimendunak) **n** eta **k**-ren balioak eskatuko dizkio erabiltzaileari.
- Gero, algoritmoa asmatze-prozesuarekin hasiko da:
  - Algoritmoak **hasierako k-CNFa sortuko du**. Algoritmoak sortutako hasierako k-CNF horri **h** deituko diogu eta formula hori ahalik eta orokorrena izango da.

Esate baterako  $k = 2$  eta  $n = 2$  kasurako sor daitezkeen formula orokorrena honako hau da:

$$x_1 \wedge (\neg x_1) \wedge x_2 \wedge (\neg x_2) \wedge (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$$

Beraz, aldagai bat eta bi aldagai erabiliz osa daitezkeen konbinazio denak,  $k = 2$  izateagatik.

$n = 5$  eta  $k = 3$  izango balira, aldagai bat, bi aldagai eta hiru aldagai erabiliz osa daitezkeen konbinazio denak hartu beharko lirateke:

$$x_1 \wedge (\neg x_1) \wedge x_2 \wedge (\neg x_2) \wedge x_3 \wedge (\neg x_3) \wedge x_4 \wedge (\neg x_4) \wedge x_5 \wedge (\neg x_5) \wedge (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee x_4) \wedge (x_1 \vee x_5) \wedge (\neg x_1 \vee x_2) \wedge \dots (\neg x_1 \vee x_2 \vee \neg x_5) \wedge \dots$$

- Algoritmoak h formula erabiltzaileari aurkeztuko dio. Erabiltzaileak g eta h baliokideak al diren erantzun behar du (Bai edo Ez erantzun behar du).
- g eta h formulak baliokideak ez badira, g True eta h False egiten duen  $v$  balorazio bat eskatuko dio algoritmoak erabiltzaileari. (Kontuz!! k-DNFetan hau alderantziz da)
- Balorazio hori kontuan hartuz algoritmoak h berri bat eraikiko du.
- Prozesua algoritmoak g-ren baliokidea den h formula bat eraiki arte errepikatuko da.

#### 6.4.2. Adibidea

- **Orain adibide bat azalduko da:**

- Erabiltzaileak  $g = (\neg x1 \vee x2)$  formula du buruan.
- $k = 2$  eta  $n = 2$  dira.
- Algoritmoak asmatze-prozesuari ekingo dio eta honako formula hau proposatuko du:  

$$h = x1 \wedge (\neg x1) \wedge x2 \wedge (\neg x2) \wedge (x1 \vee x2) \wedge (x1 \vee \neg x2) \wedge (\neg x1 \vee x2) \wedge (\neg x1 \vee \neg x2)$$
- Algoritmoak g eta h baliokideak al diren galdetuko dio erabiltzaileari. Ez direnez baliokideak, ezetz esateaz gain, g True eta h False egiten duen  $v$  balorazio bat eman behar du erabiltzaileak:  

$$v = (\text{False}, \text{False})$$
- $v$  balorazioarekin False egiten diren klausulak ezabatuko ditu algoritmoak:  

$$h = \cancel{x1} \wedge (\neg \cancel{x1}) \wedge \cancel{x2} \wedge (\neg \cancel{x2}) \wedge (\cancel{x1} \vee \cancel{x2}) \wedge (x1 \vee \neg x2) \wedge (\neg x1 \vee x2) \wedge (\neg x1 \vee \neg x2)$$
  
 Beraz, h honela geldituko da:  

$$h = (\neg x1) \wedge (\neg x2) \wedge (x1 \vee \neg x2) \wedge (\neg x1 \vee x2) \wedge (\neg x1 \vee \neg x2)$$
- Algoritmoak g eta h baliokideak al diren galdetuko dio erabiltzaileari. Ez direnez baliokideak, ezetz esateaz gain, g True eta h False egiten duen  $v$  balorazio bat eman behar du erabiltzaileak:  

$$v = (\text{True}, \text{True})$$
- Balorazio berri hau kontuan hartuz, False egiten diren klausulak ezabatuko ditu algoritmoak:  

$$h = (\neg \cancel{x1}) \wedge (\neg \cancel{x2}) \wedge (x1 \vee \neg x2) \wedge (\neg x1 \vee x2) \wedge (\neg \cancel{x1} \vee \neg \cancel{x2})$$

Beraz, h berria honako hau izango da:

$$h = (x1 \vee \neg x2) \wedge (\neg x1 \vee x2)$$

- Algoritmoak g eta h baliokideak al diren galdetuko dio erabiltzaileari. Ez direnez baliokideak, ezetz esateaz gain, g True eta h False egiten duen v balorazio bat eman behar du erabiltzaileak:

$$v = (\text{False}, \text{True})$$

- Balorazio berri hau kontuan hartuz, False egiten diren klausulak ezabatuko ditu algoritmoak:

$$h = (\cancel{x1} \vee \cancel{\neg x2}) \wedge (\neg x1 \vee x2)$$

Beraz, h berria honako hau izango da:

$$h = (\neg x1 \vee x2)$$

- Algoritmoak g eta h baliokideak al diren galdetuko dio erabiltzaileari. Baliokideak direnez, erabiltzaileak baietz erantzungo du eta prozesua bukatu egingo da. Algoritmoak formula asmatu egin du.

### **6.4.3. Algoritmoa**

- **Algoritmoa honako hau da:**

$$\{n \geq k \wedge k\_cnf\_a\_da(g)\}$$

h := hasierako\_k\_cnf\_a(n, k);

baliokid := erabiltzaileari\_formula\_baliokidea\_al\_den\_galdetu(h);

**while** not baliokid **loop**

v := erabiltzaileari\_balorazioa\_eskatu ();

klausulak\_ezabatu(h, v);

baliokid:= erabiltzaileari\_formula\_baliokidea\_al\_den\_galdetu(h);

**end loop**

$$\{k\_cnf\_a\_da(h) \wedge h \leftrightarrow g\}$$

### **6.4.4. Algoritmoak erabiltzaileari egin diezazkiokeen galdera motak**

Algoritmo honek bi eratako galderak egiten dizkio erabiltzaileari:

- ✓ g True eta h False egiten dituen v balorazio bat eskatu.
- ✓ h eta g baliokideak al diren galdetu.

### 6.5. k-DNFak

Formula boolear bat k-DNFa dela esaten da literalen konjuntzioak diren azpiformulen disjuntzio bat baldin bada eta konjuntzio bakoitzak gehienez k literal baldin baditu. Literal bat aldagai bat edo ukatutako aldagai bat da.

Honako formula hau 3-DNFa da:

$$(\neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_4) \vee (x_2 \wedge x_3) \vee (\neg x_1 \wedge \neg x_4)$$

#### 6.5.1. Egoera

- Erabiltzaileak k-DNF bat du buruan (formula horri g deituko diogu). Formula horretan  $x_1, x_2, \dots, x_n$  aldagaiak ager daitezke. Hala ere, garbi eduki behar da gerta daitekeela  $x_1, x_2, \dots, x_n$  tarteko aldagairen batzuk ez agertzea g formularen. Bestalde  $1 \leq k \leq n$  beteko da beti.

Esate baterako  $k = 2$  baldin bada eta  $n = 4$  baldin bada, erabiltzaileak honako formula hau eduki dezake buruan:

$$g = (\neg x_3) \vee (x_2 \wedge x_3) \vee (x_4)$$

Formula horretan  $x_1$  ez da agertzen.

- Hasteko, algoritmoak (Sistema adimendunak) **n eta k-ren balioak eskatuko** dizkio erabiltzaileari.
- Gero, algoritmoa asmatze-prozesuarekin hasiko da:
  - Algoritmoak **hasierako k-DNFa sortuko du**. Algoritmoak sortutako hasierako k-DNF horri **h** deituko diogu eta formula hori ahalik eta orokorrena izango da.

Esate baterako  $k = 2$  eta  $n = 2$  kasurako sor daitezkeen formula orokorrena honako hau da:

$$x_1 \vee (\neg x_1) \vee x_2 \vee (\neg x_2) \vee (x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2)$$

Beraz, aldagai bat eta bi aldagai erabiliz osa daitezkeen konbinazio denak, k = 2 izateagatik.

$n = 5$  eta  $k = 3$  izango balira, aldagai bat, bi aldagai eta hiru aldagai erabiliz osa daitezkeen konbinazio denak hartu beharko lirateke:

$$x_1 \vee (\neg x_1) \vee x_2 \vee (\neg x_2) \vee x_3 \vee (\neg x_3) \vee x_4 \vee (\neg x_4) \vee x_5 \vee (\neg x_5) \vee (x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_1 \wedge x_4) \vee (x_1 \wedge x_5) \vee (\neg x_1 \wedge x_2) \vee \dots \vee (\neg x_1 \wedge x_2 \wedge \neg x_5) \vee \dots$$

- Algoritmoak h formula erabiltzaileari aurkeztuko dio. Erabiltzaileak g eta h baliokideak al diren erantzun behar du (Bai edo Ez erantzun behar du).
- g eta h formulak baliokideak ez badira, g False eta h True egiten duen v balorazio bat eskatuko dio algoritmoak erabiltzaileari. (Kontuz!! k-CNFetan hau alderantziz da)
- Balorazio hori kontuan hartuz algoritmoak h berri bat eraikiko du.
- Prozesua algoritmoak g-ren baliokidea den h formula bat eraiki arte errepikatuko da.

### 6.5.2. Adibidea

- **Orain adibide bat azalduko da:**

- Erabiltzaileak  $g = (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)$  formula du buruan.
- $k = 2$  eta  $n = 2$  dira.
- Algoritmoak asmatze-prozesuari ekingo dio eta honako formula hau proposatuko du:  

$$h = x_1 \vee (\neg x_1) \vee x_2 \vee (\neg x_2) \vee (x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2)$$
- Algoritmoak g eta h baliokideak al diren galdetuko dio erabiltzaileari. Ez direnez baliokideak, ezetz esateaz gain, g False eta h True egiten duen v balorazio bat eman behar du erabiltzaileak:  

$$v = (\text{False}, \text{False})$$
- v balorazioarekin True egiten diren terminoak ezabatuko ditu algoritmoak:  

$$h = x_1 \vee (\neg x_1) \vee x_2 \vee (\neg x_2) \vee (x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2)$$
  

$$h \text{ berria honako hau izango da:}$$

$$h = x_1 \vee x_2 \vee (x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)$$
- Algoritmoak g eta h baliokideak al diren galdetuko dio erabiltzaileari. Ez direnez baliokideak, ezetz esateaz gain, g False eta h True egiten duen v balorazio bat eman behar du erabiltzaileak:  

$$v = (\text{True}, \text{True})$$
- v balorazioarekin True egiten diren terminoak ezabatuko ditu algoritmoak:  

$$h = x_1 \vee x_2 \vee (x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)$$
  

$$h \text{ berria honako hau izango da:}$$

$$h = (x1 \wedge \neg x2) \vee (\neg x1 \wedge x2)$$

- Algoritmoak g eta h baliokideak al diren galdetuko dio erabiltzaileari. Baliokideak direnez, erabiltzaileak baietz erantzungo du eta prozesua bukatu egingo da. Algoritmoak formula asmatu egin du.

### **6.5.3. Algoritmoa**

- **Algoritmoa honako hau da:**

$\{n \geq k \wedge k\_dnf\_a\_da(g)\}$

$h := \text{hasierako\_k\_dnf\_a}(n, k);$

$\text{baliokid} := \text{erabiltzaileari\_formula\_baliokidea\_al\_den\_galdetu}(h);$

**while** not baliokid **loop**

$v := \text{erabiltzaileari\_balorazioa\_eskatu}();$

$\text{terminoak\_ezabatu}(h, v);$

$\text{baliokid} := \text{erabiltzaileari\_formula\_baliokidea\_al\_den\_galdetu}(h);$

**end loop**

$\{k\_dnf\_a\_da(h) \wedge h \leftrightarrow g\}$

### **6.5.4. Algoritmoak erabiltzaileari egin diezazkiokeen galdera motak**

Algoritmo honek bi eratako galderak egiten dizkio erabiltzaileari:

- ✓ g False eta h True egiten dituen v balorazio bat eskatu.
- ✓ h eta g baliokideak al diren galdetu.