Exercițiul 1

Subpunctul a

Inițial, mulțimea de ecuații pe care trebuie să o rezolvăm conține

$$T = \{\, f(h(h(g(c, X))), c, g(h(X), Z)) \doteq f(h(Z), X, Y)\,\}$$

Mulțimea de substituții este vidă $S = \emptyset$.

Algoritm: Extragem pe rând câte o egalitate din mulțime, o rezolvăm aplicând regulile de unificare și continuăm până mulțimea T devine vidă.

- Unificăm

  $$f(h(h(g(c, X))), c, g(h(X), Z)) \doteq f(h(Z), X, Y)$$

  Vedem că acești termeni sunt o aplicare a funcției f:

  $$f(\dots) \doteq f(\dots)$$

  Eliminăm f și inserăm în mulțimea T egalitățile obținute egalând parametrii:

  $$T = \{\, h(h(g(c, X))) \doteq h(Z), c \doteq X, Z \doteq Y\,\}$$

- Unificăm

  $$h(h(g(c, X))) \doteq h(Z)$$

  Din nou, observăm că este o aplicare a funcției h. O eliminăm și inserăm egalități în mulțime:

  $$T = \{\, c \doteq X, Z \doteq Y, h(g(c, X)) \doteq Z\,\}$$

- Unificăm

  $$c \doteq X$$

  Fiind vorba de egalitate între o constantă și o variabilă, putem să eliminăm această egalitate și să aplicăm peste tot substituția $X \leftarrow c$. Mulțimea de egalități devine

  $$T = \{\, Z \doteq Y, h(g(c, c)) \doteq Z\,\}$$

  și mulțimea de substituții devine

  $$S = \{\, X \leftarrow c\,\}$$

- Unificăm
$$Z \doteq Y$$
Fiind vorba de egalitate de variabile, putem să o eliminăm și să aplicăm substituția $Y \leftarrow Z$. Mulțimea de egalități devine
$$T = \{\, h(g(c,c)) \doteq Z \,\}$$
și mulțimea de substituții devine
$$S = \{\, X \leftarrow c, Y \leftarrow Z \,\}$$

- Unificăm
$$h(g(c,c)) \doteq Z$$
Fiind vorba de egalitate între o variabilă și un termen compus, trebuie să facem substituția $Z \leftarrow h(g(c,c))$. Mulțimea de egalități devine $T = \emptyset$ și mulțimea finală de substituții este
$$S = \{\, X \leftarrow c, Y \leftarrow Z, Z \leftarrow h(g(c,c)) \,\}$$
Cel mai general unificator pentru termenii inițiali este:
$$cgu = \{\, X \leftarrow c, Y \leftarrow h(g(c,c)), Z \leftarrow h(g(c,c)) \,\}$$

Subpunctul b

Începem prin a aduce enunțul în formă prenex, folosind echivalențe:

$$\begin{aligned}
&\forall X(\exists Y p(X,h(Y)) \lor \exists Z(q(f(a,X,h(Z))) \land r(h(Z))))\\
\iff &\forall X \exists Y(p(X,h(Y)) \lor \exists Z(q(f(a,X,h(Z))) \land r(h(Z))))\\
\iff &\forall X \exists Y \exists Z(p(X,h(Y)) \lor (q(f(a,X,h(Z))) \land r(h(Z))))
\end{aligned}$$

Înlocuim fiecare variabilă cuantificată existențial cu o funcție care depinde de variabilele cuantificate universal de dinaintea ei.

Înlocuim Z cu $f_Z(X)$:
$$\forall X \exists Y(p(X,h(Y)) \lor (q(f(a,X,h(f_Z(X)))) \land r(h(f_Z(X)))))$$
Înlocuim Y cu $f_Y(X)$:
$$\forall X(p(X,h(f_Y(X))) \lor (q(f(a,X,h(f_Z(X)))) \land r(h(f_Z(X)))))$$

Putem elimina cuantificatorii, pentru că știm că toate variabilele care apar sunt cuantificate universal:

$$p(X, h(f_Y(X))) \lor (q(f(a, X, h(f_Z(X)))) \land r(h(f_Z(X))))$$

În acest moment, enunțul se află în **formă Skolem**. Pentru a aplica Davis-Putnam, trebuie să îl aducem la o **formă clauzală**.

Trebuie să aducem enunțul la formă normală conjunctivă, folosindu-ne de echivalențe:

$$\begin{aligned} & p(X, h(f_Y(X))) \lor (q(f(a, X, h(f_Z(X)))) \land r(h(f_Z(X)))) \\ \iff & (p(X, h(f_Y(X))) \lor q(f(a, X, h(f_Z(X)))))\land \\ & (p(X, h(f_Y(X))) \lor r(h(f_Z(X)))) \end{aligned}$$

Fiecare disjuncție din FNC devine o mulțime de termeni (o **clauză**). Acestea sunt puse într-o singură mulțime:

$$\{\{\, p(X, h(f_Y(X))), q(f(a, X, h(f_Z(X))))\,\}, \\ \{\, p(X, h(f_Y(X))), r(h(f_Z(X)))\,\}\}$$

Acum ar trebui să aplicăm rezoluția din logica de ordinul I. Ar trebui să avem un termen care apare normal într-o clauză și în alta negat.

Neavând nicio negație, algoritmul Davis-Putnam se oprește imediat.

Enunțul este satisfiabil.

Exercițiul 2

Pentru a demonstra o propoziție prin derivare, negăm ținta și încercăm să ajungem la clauza vidă prin **rezoluție**.

La început, arborele conține doar ținta negată:

$$\neg consuma(Cine, Ce)$$

Observăm că putem aplica rezoluția între acest termen și

- $consuma(ana, X)$, unificând prin substituțiile $Cine \leftarrow ana$, $Ce \leftarrow X$
- $consuma(victor, X)$, unificând prin $Cine \leftarrow victor$, $Ce \leftarrow X$

Arborele rezultat este:

$$\neg consuma(Cine, Ce)$$

$$\neg consuma(ana, X) \qquad \neg consuma(victor, X)$$

Luăm ca nouă țintă $\neg consuma(ana, X)$, și încercăm să facem rezoluția cu ce avem în baza de date. Avem un singur predicat care se potrivește. Obținem

$$\neg consuma(ana, X)$$

$$\neg fructe(X) \vee \neg sunt(X, rosii)$$

Aplicăm rezoluția pe această nouă țintă cu $fructe(mere)$, $fructe(pere)$ și $fructe(capsuni)$, și obținem:

$$\neg fructe(X) \vee \neg sunt(X, rosii)$$

$\neg sunt(mere, rosii)$ (cu $X \leftarrow mere$) | $\neg sunt(pere, rosii)$ (cu $X \leftarrow pere$) | $\neg sunt(capsuni, rosii)$ (cu $X \leftarrow capsuni$)

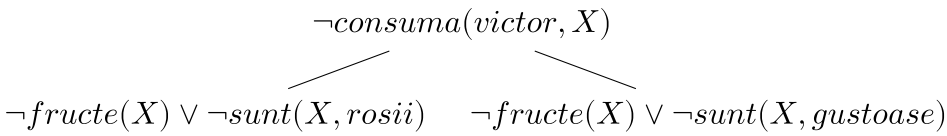
Aplicăm încă o dată rezoluția și obținem câteva soluții:

$$\neg fructe(X) \vee \neg sunt(X, rosii)$$

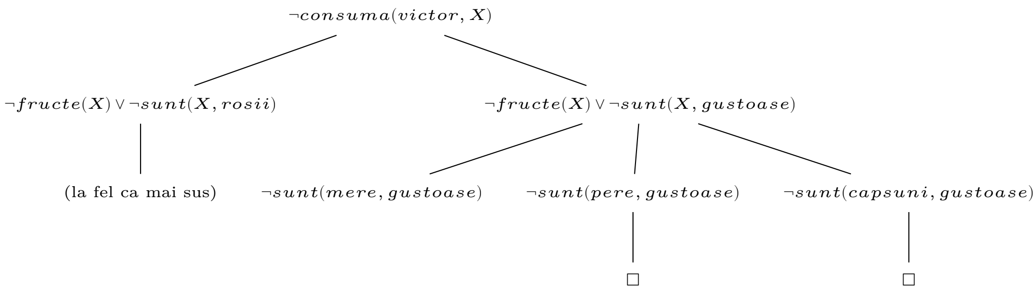
$\neg sunt(mere, rosii)$ | $\neg sunt(pere, rosii)$ (nu avem cu ce să facem rezoluție) | $\neg sunt(capsuni, rosii)$

$\square$ (rezoluție cu $sunt(mere, rosii)$) | | $\square$ (rezoluție cu $sunt(capsuni, rosii)$)

Ne întoarcem, facem la fel pentru ținta $\neg consuma(victor, X)$ (aici avem mai multe clauze cu care putem face rezoluție):



Arborele pentru $\neg consuma(victor, X)$ devine:



Exercițiul 3

Codul sursă pe GitHub

Exercițiul 4

Codul sursă pe GitHub