Sugestii Rezolvări Exerciții Decidabilitate

October 2018

- **Exercițiul 1**
  - $A \leq_T B$ si B semidecidabilă $\Rightarrow A$ semidecidabilă.
    Explicație:
    B semidecidabilă $\Rightarrow$ există procedura P_B, care, pentru o intrare oarecare x, întoarce 1 când $x \in B$ și nu oferă niciun răspuns atunci când $x \notin B$.
    $A \leq_T B \Rightarrow$ orice intrare x pentru predicatul P_A care decide apartenența la mulțimea A poate fi transformat într-o intrare $f(x)$ pentru P_B astfel încât $P_A(x) = 1 \iff P_B(f(x)) = 1$
    Cu alte cuvinte, putem folosi P_B pentru a stabili ca un element oarecare x aparține lui $A \Rightarrow$ există o procedură P_A care, pentru o intrare oarecare x, întoarce 1 când $x \in A$ și nu oferă niciun răspuns atunci când $x \notin A \Rightarrow A$ semidecidabilă.
    În particular, A ar putea fi chiar decidabilă, însă acest lucru nu rezultă din simpla relație $A \leq_T B$.
  - A semidecidabilă și C decidabilă $\Rightarrow A \setminus C$ semidecidabilă.
    Explicație:
    C decidabilă $\Rightarrow$ există procedura P_C astfel încât $P_C(x) = 1$, pentru $x \in C$ și $P_C(x) = 0$, pentru $x \notin C$.
    Pentru a decide ca $x \in A \setminus C$ vom rula $P_A(x)$ și în cazul în care răspunsul este 1, vom cere și ca $P_C(x) = 0$. Dacă $P_A(x)$ nu oferă niciun raspuns, înseamnă că $x \notin A$, deci $x \notin A \setminus C$. Prin urmare, putem construi o procedură care, pentru o intrare oarecare x, întoarce 1 atunci când $x \in A \setminus C$ și nu oferă niciun răspuns în caz contrar $\Rightarrow A \setminus C$ semidecidabila.
- **Exercițiul 3**

  Hint: De ce este varianta PCP cu alfabet unar decidabilă?
- **Exercițiul 4**

  Hint: Mulțimea A este finită.
- **Exercițiul 6**

  Hint: Pentru a demonstra că o mulțime A este recursiv-numărabilă este suficient să arătăm că există un program generator pentru mulțime.
- **Exercițiul 10**

  Din (c) $\Rightarrow$ există P_A, P_B, P_C programele care decid mulțimile respective (întorc 1 dacă elementul aparține mulțimii, altfel nu se termină).

Din B recursiv-numărabilă și C recursiv-numărabilă $\Rightarrow B \cup C$ recursiv-numărabilă (se poate scrie un program $P_{B \cup C}(x)$ care să ruleze în paralel $P_B(x)$ și $P_C(x)$ și care se termină când unul din cele doua programe se termină.
Dar $B \cup C = N \setminus A$ pentru că A, B, C sunt disjuncte între ele (din (a)) și reuniunea lor este mulțimea numerelor naturale (din (b)).
Din A recursiv-numarabilă și $N \setminus A$ recursiv numărabilă $\Rightarrow A$ recursivă (se poate construi programul $P'_A(x)$ care rulează în paralel $P_A(x)$ si $P_{N \setminus A}(x)$; daca $P_A(x)$ se termină primul, atunci $x \in A$, deci P'_A întoarce 1; altfel, dacă $P_{N \setminus A}$ se termină primul, atunci $x \in N \setminus A$ deci P'_A întoarce 0. **QED**
Analog pentru mulțimile B si C bazându-ne pe proprietățile de asociativitate și comutativitate ale operației de reuniune.

- **Exercițiul 11**

  Problema BIN se aseamana cu problema opririi(PO). Prin urmare, dorim sa demonstrăm că BIN este nedecidabila, prin demonstrarea reducerii $PO \leq_T BIN$.

  - **Pasul 1:** Pentru o intrare oarecare (P, w) pentru PO, construim o intrare convenabilă (P', w') a lui BIN, astfel încât $PO(P, w) = 1 \iff BIN(P', w') = 1$.

```
function P'(w')
    P(w)
    if w' is in {0, 1}*
        return 1
    else
        infinite-loop
end function
```

  - **Pasul 2:** $PO(P, w) = 1 \Rightarrow BIN(P', w') = 1$. Când P se oprește pe w (adica $PO(P, w) = 1$), P' se va opri și el, pentru orice intrare w' care are proprietatea din enunț.
  - **Pasul 3:** $BIN(P', w') = 1 \Rightarrow PO(P, w) = 1$. Când există inputuri pe care P' se oprește, înseamnă că acele intrări sunt în format binar și că $P(w)$ s-a oprit.