

**7.7** Stel  $A$  en  $B$  in NP. Te bewijzen:  $A \cup B$  en  $A \cdot B$  in NP. Zij  $V_A$  en  $V_B$  polynomiale verifiers voor  $A$  en  $B$ .

Volgend algoritme is een polynomiale verifier voor  $A \cup B$ : “Op input  $(x, c)$ , roep  $V_A(x, c)$  en  $V_B(x, c)$  aan, en aanvaard als minstens een van deze twee aanvaardt.”

Volgend algoritme is een polynomiale verifier voor  $A \cdot B$ : “Op input  $(x, c)$ , controleer eerst of  $c$  van de vorm  $(i, c_A, c_B)$  is, waarbij  $i \in \{0, 1, \dots, n\}$  ( $n$  is zoals altijd de lengte van input  $x$ ). Roep nu  $V_A(x_1 \dots x_i, c_A)$  en  $V_B(x_{i+1} \dots x_n, c_B)$  aan, en aanvaard als beiden aanvaarden.”

**7.14** Stel  $A$  in NP, en zij  $V$  een polynomiale verifier voor  $A$ . Volgende polynomiale verifier voor  $A^*$  toont aan dat ook  $A^*$  in NP zit: “Op input  $(x, c)$ , controleer eerst of  $c$  van de vorm  $(x_1, c_1, x_2, c_2, \dots, x_m, c_m)$  is, waarbij  $x_1 x_2 \dots x_m = x$ . Roep dan  $V$  aan voor de  $m$  verschillende inputs  $(x_1, c_1)$ ,  $(x_2, c_2)$ ,  $\dots$ ,  $(x_m, c_m)$ , en aanvaard als al deze aanroepen aanvaarden.”

**LPATH in NP** Hier is een polynomiale verifier: “Op input  $(\langle G, a, b, k \rangle, c)$ , controleer of  $c$  een enkelvoudig pad is van  $a$  naar  $b$  van lengte hoogstens  $k$ .”

**SPATH in P** Run een breadth-first search vanuit  $a$  voor  $k$  iteraties. We hebben dan alle knopen bepaald die bereikbaar zijn vanuit  $a$  met een enkelvoudig pad van hoogstens  $k$  pijlen. Aanvaard als  $b$  daar tussen zit.

**DOUBLE-SAT in NP** Hier is een polynomiale verifier: “Op input  $(\langle \phi \rangle, c)$ , controleer eerst of  $c$  bestaat uit twee assignments van de variabelen in  $\phi$ . Controleer dan of ze verschillend zijn (dus verschillen in de waarde toegekend aan minstens één variabele), en of  $\phi$  geëvalueerd onder beide assignments true wordt.”

**HALF-CLIQUE in NP** Hier is een polynomiale verifier: “Op input  $(\langle G \rangle, c)$ , controleer of  $c$  bestaat uit een lijst knopen van  $G$ ; noem de verzameling van de knopen in deze lijst  $A$ . Tel nu het aantal pijlen vanuit een knoop in  $A$  naar een knoop niet in  $A$ . Aanvaard indien dit aantal minstens  $k$  is.”

**MAX-CUT in NP** Hier is een polynomiale verifier: “Op input  $(\langle G, k \rangle, c)$ , controleer eerst of  $c$  bestaat uit een lijst knopen van  $G$ ; noem de verzameling van de knopen in deze lijst  $A$ . Tel nu het aantal pijlen vanuit een knoop in  $A$  naar een knoop niet in  $A$ . Aanvaard indien dit aantal minstens  $k$  is.”