Zadanie 1. Oto gramatyka pewnego języka programowania:

$$
\begin{array}{lll}
e & ::= & N \mid \\
 & & x \mid \\
 & & e_1 + e_2 \mid \\
 & & e_1 - e_2 \\
b & ::= & true \mid \\
 & & b_1 \wedge b_2 \mid \\
 & & \neg b \mid \\
 & & e_1 > e_2 \\
gcom & ::= & b \rightarrow i \mid \\
 & & b \rightarrow i \; [\!] \; gcom \\
i & ::= & \textbf{skip} \mid \\
 & & \textbf{abort} \mid \\
 & & x := e \mid \\
 & & i_1 ; i_2 \mid \\
 & & \textbf{if } gcom \textbf{ fi} \mid \\
 & & \textbf{do } gcom \textbf{ od}
\end{array}
$$

Wyrażenia arytmetyczne i boolowskie oraz instrukcje **skip**, przypisanie i złożenie mają znaczenie standardowe. Instrukcja **abort** powoduje zakończenie programu w stanie *porażki*.

Instrukcje **if** i **do** są niedeterministyczne. Składają się z gałęzi - instrukcji dozorowanych, z których każda składa sie z warunku b i instrukcji i. Warunki b nie muszą się wzajemnie wykluczać.

Wykonanie instrukcji **if** polega na niedeterministycznym wyborze instrukcji dozorowanej spośród tych, których warunki b są spełnione, i wykonaniu jej instrukcji i. Jeśli wszystkie warunki są fałszywe, instrukcja **if** jest równoważna instrukcji **abort**.

Instrukcja **do** jest pętlą. Jej wykonanie polega na niedeterministycznym wyborze instrukcji dozorowanej spośród tych, których warunki b są spełnione, wykonaniu jej instrukcji i, a następnie ponownym wykonaniu pętli w stanie wynikowym wykonanej instrukcji. Jeśli wszystkie warunki są fałszywe, wykonanie pętli kończy się.

Zdefiniuj semantykę naturalną ("duże kroki") tego języka. Wystarczy podać opis konfiguracji i reguły dla instrukcji (także dozorowanych) przyjmując, że dana jest semantyka dla wyrażeń i wyrażeń logicznych, definiująca dla dowolnego wyrażenia e i stanu s relację $(e, s) \rightarrow n$, gdzie n jest (zawsze określoną) wartością wyrażenia e w stanie s, a dla wyrażeń logicznych b, relację $(b, s) \rightarrow w$, gdzie $w \in \{tt, ff\}$ jest (zawsze określoną) wartością wyrażenia b w stanie s.

Przykład programu w tym języku:

$$
\begin{array}{l}
\textbf{if} \quad x > 0 \wedge y > 0 \wedge z > 0 \rightarrow \\
\quad \textbf{do} \\
\qquad x > y \rightarrow x := x - y \; [\!] \\
\qquad y > z \rightarrow y := y - z \; [\!] \\
\qquad z > x \rightarrow z := z - x \; [\!] \\
\quad \textbf{od} \\
\textbf{fi}
\end{array}
$$

Program rozpoczęty w stanie (x=12, y=4, z=2) zakończy się w stanie (x=2, y=2, z=2) czyli NDW(x,y,z).
Program rozpoczęty w stanie (x=0, y=-1, z=2) zakończy się w stanie porażki.