Zadanie 3. Niech dany będzie deterministyczny VASS $\mathcal{A}$ o d licznikach. Niech $f(i)$ oznacza wartość i-tego licznika w konfiguracji akceptującej. Możliwe jest pięć sytuacji, w których $\mathcal{A}$ nie akceptuje słowa.

A Automat zawiesza się, bo nie ma przejścia z aktualnego stanu po wczytanej literze.

B Automat zawiesza się, bo przejście po wczytanej literze spowodowałoby spadek któregoś licznika poniżej 0.

C Automat wczytuje całe słowo, ale zostaje w stanie, który nie jest akceptujący.

D Automat wczytuje całe słowo, ale dla pewnego i wartość i-tego licznika na końcu wynosi mniej niż $f(i)$.

E Automat wczytuje całe słowo, ale dla pewnego i wartość i-tego licznika na końcu wynosi więcej niż $f(i)$.

Stworzymy kilka komponentów sprawdzających, czy powyższe sytuacje zachodzą. Ich połączenie będzie stanowić automat z jednym licznikiem, którego językiem jest dokładnie dopełnienie języka rozpoznawanego przez $\mathcal{A}$. Stwórzmy najpierw nowy stan F oraz ustalmy, że jedyną konfiguracją akceptującą będzie $(F, 0)$. Dodajmy też stan stan F' oraz tranzycje z F' do F i z F' do F' po każdej literze alfabetu (nie zmieniające licznika).

A Tworzymy kopię A automatu $\mathcal{A}$ ale z jednym licznikiem, który będzie stale równy 0. To znaczy stany są takie jak w $\mathcal{A}$ oraz tranzycje takie jak w $\mathcal{A}$, ale nie zmieniające licznika. Stan początkowy jest jak w $\mathcal{A}$. Teraz dodatkowo, jeśli ze stanu s w A nie prowadzi żadna tranzycja po literze a z alfabetu, to dodajemy tranzycję po literze a ze stanu s do F oraz tranzycję po literze a ze stanu s do F' (dla sytuacji, gdy zostało do wczytania jeszcze więcej liter). Zauważmy, że automat A akceptuje słowo wtedy i tylko wtedy, gdy $\mathcal{A}$ zawiesza się z powodu braku tranzycji.

B Dla każdego $i = 1, \ldots, d$ tworzymy automat B_i z jednym licznikiem, który jest kopią automatu $\mathcal{A}$, ale tylko z licznikiem o numerze i (to znaczy stany są takie same oraz tranzycje są takie same, ale zmieniają licznik automatu dokładnie tak jak i-ty licznik $\mathcal{A}$). Dla każdego B_i

dodajemy jeszcze jeden stan p, który będzie jedyny początkowy w B_i. Tranzycje wychodzące z p mają być kopiami tranzycji wychodzących ze stanu początkowego $\mathcal{A}$, ale jeśli taka tranzycja zwiększała i-ty licznik o w, to kopia wychodząca z p ma zwiększać licznik o $w+1$.

Dla każdej tranzycji w B_i z s_1 do s_2 dodającej do licznika $w < 0$, dodajemy $|w|$ nowych tranzycji z s_1 do s_2 dodających do licznika $w+1, w+2, \ldots, 0$.

Dodatkowo, teraz dla każdej tranzycji w automacie B_i z s_1 do s_2 po literze a, która dodaje w do licznika, dodajemy też tranzycje z s_1 do F i z s_1 do F', które są po literze a oraz dodają do licznika w.

Tak skonstruowany komponent akceptuje słowo wtedy i tylko wtedy, gdy podczas wczytywania słowa, dla pewnego i, i-ty licznik automatu $\mathcal{A}$ spadł poniżej 0. Istotnie, jeśli tak było, to w komponentcie B_i możliwe jest używanie tranzycji, które będą zmniejszać licznik o dowolną mniejszą wartość, więc będzie możliwe skończenie w stanie F z wartością licznika 0, aby w żadnym kroku licznik nie spadł poniżej 0. Z drugiej strony, jeśli i-ty licznik automatu $\mathcal{A}$ podczas wczytywania słowa jest zawsze nieujemny, to podczas wczytywania tego słowa wartość licznika B_i będzie w każdym kroku wynosiła co najmniej 1 (bo po przejściu ze stanu p zwiększyliśmy licznik o 1 więcej niż automat $\mathcal{A}$).

C Tworzymy kopię C automatu $\mathcal{A}$, która będzie miała jeden licznik stale równy 0 (tak jak w A). Stan początkowy odpowiada stanowi początkowemu z $\mathcal{A}$. Jeśli s jest stanem, który odpowiada stanowi różnemu od akceptującego z $\mathcal{A}$, to dla każdej tranzycji prowadzącej do s, dodajemy kopię tej tranzycji prowadzącą do F. Zauważmy, że automat C akceptuje wtedy i tylko wtedy, gdy $\mathcal{A}$ kończy w stanie różnym od akceptującego, o ile wczyta całe słowo.

D Dla każdego $i = 1, \ldots, d$ oraz każdego $j = 0, 1, \ldots, f(i) - 1$ tworzymy kopię D_{ij} automatu $\mathcal{A}$ tylko z i-tym licznikiem (podobnie jak w B). Stany początkowe odpowiadają stanowi początkowemu z $\mathcal{A}$. Tym razem jedyna modyfikacja będzie taka, że jeśli f odpowiada stanowi akceptującemu z $\mathcal{A}$ w D_{ij}, to dla każdej tranzycji prowadzącej do f i dodającej do licznika w, tworzymy jej kopię prowadzącą do F i dodającą do licznika $w - j$. Dla ustalonego i, komponent złożony z automatów D_{ij} akceptuje słowo, wtedy i tylko wtedy, gdy automat $\mathcal{A}$ wczytał to słowo oraz stan i-tego licznika na koniec wyniósł mniej niż $f(i)$.

E Dla każdego $i = 1, \ldots, d$ tworzymy kopię E_i automatu $\mathcal{A}$ tylko z i-tym licznikiem (podobnie jak w B). Stany początkowe odpowiadają stanowi początkowemu z $\mathcal{A}$.

Dla każdej tranzycji w E_i z s_1 do s_2 dodającej do licznika $w > 0$, dodajemy $|w|$ nowych tranzycji z s_1 do s_2 dodających do licznika $w - 1, w - 2, \ldots, 0$. Co więcej, jeśli f odpowiada stanowi akceptującemu z $\mathcal{A}$ w E_i, to dla każdej tranzycji prowadzącej do f i dodającej do licznika w, tworzymy jej kopię prowadzącą do F i dodającą do licznika $w - f(i) - 1$.

Komponent złożony z automatów E_i akceptuje słowo, wtedy i tylko wtedy, gdy automat $\mathcal{A}$ wczytał to słowo oraz stan i-tego licznika na koniec wyniósł więcej niż $f(i)$. Wynika to z faktu, że w E_i ostatnia tranzycja odejmuje jeszcze $f(i) + 1$. Z drugiej strony nowe tranzycje pozwalają na zwiększanie licznika tak, aby nie był „zbyt duży"(aby możliwe było otrzymanie zerowej wartości licznika na koniec).

Z połączenia komponentów z powyższych podpunktów powstaje więc automat z jednym licznikiem, który rozpoznaje dopełnienie języka $\mathcal{A}$ (niedeterministyczny, ale bez ε-przejść).