2. Wykład 2: Indukcja

2.1. **Definiowanie kategorii syntaktycznych przez indukcję.** Istotnym składnikiem definicji języka programowania jest jego **składnia syntaktyczna,** dzięki której jesteśmy w stanie odpowiedzieć jaki program (rozumiany jako ciąg znaków) będzie rozpoznany przez parser, a jaki nie będzie. Na ogół składnia syntaktyczna jest określona za pomocą pewnej liczby **kategorii syntaktycznych** takich jak wyrażenia, typy i wzorce. Poniżej przedstawimy na kilku przykładach w jaki sposób można definiować kategorie syntaktyczne przez indukcję w kilku prostych językach.

Jako pierwszy przykład rozważmy następującą definicję kategorii syntaktycznej nat liczb naturalnych:

$$\mathsf{nat} \qquad n \ ::= \ \mathtt{0} \mid \mathtt{S}\ n$$

W przykładzie tym nat jest nazwą kategorii syntaktycznej, którą definiujemy, n jest zwany **nie-terminalem,** ::= czytamy *"jest definiowane jako"*, a I jako *"lub"*, 0 oznacza *"zero"*, zaś S oznacza *"następnik"*. Tym samym definicję można odczytać następująco:

liczba naturalna n jest definiowana jako 0 albo S n', gdzie n' jest inną liczbą naturalną.

Zauważmy, że nat jest tu definiowane indukcyjnie: liczba naturalna S n' jest zdefiniowana za pomocą innej liczby naturalnej n' i tym samym nat używa tej samej kategorii syntaktycznej w swej definicji. A zatem definicja nat produkuje nieskończony zbiór liczb naturalnych takich jak

$$\mathtt{0}, \mathtt{S}\ \mathtt{0}, \mathtt{S}\ \mathtt{S}\ \mathtt{0}, \mathtt{S}\ \mathtt{S}\ \mathtt{S}\ \mathtt{0}, \ldots$$

Kategoria syntaktyczna może odnosić się w swej definicji do innej kategorii syntaktycznej. Na przykład, mając daną zdefiniowaną powyżej kategorię nat, możemy indukcyjnie zdefiniować nową kategorię syntaktyczną tree jak następuje:

$$\mathsf{tree} \qquad t \ ::= \ \mathtt{leaf}\ n \mid \mathtt{node}(t, n, t)$$

Tu z kolei leaf n oznacza liść drzewa o numerze n, node (t_1, n, t_2) oznacza węzeł o numerze n z lewym potomkiem t_1 i prawym potomkiem t_2. Zatem tree specyfikuje język drzewek binarnych liczb naturalnych takich jak

$$\mathtt{leaf}\ n, \mathtt{node}\ (\mathtt{leaf}\ n_1, n, \mathtt{leaf}\ n_2), \mathtt{node}\ (\mathtt{node}\ (\mathtt{leaf}\ n_1, n, \mathtt{leaf}\ n_2), n', n''), \ldots$$

W podobny sposób możemy pokusić się o zdefiniowanie dwóch kategorii syntaktycznych które wzajemnie definiują się przez indukcję. Na przykład możemy równocześnie zdefiniować kategorie par i npar liczb parzystych i nieparzystych jak następuje

$$\begin{array}{ll} \mathsf{par} & e \ ::= \ \mathtt{0} \mid \mathtt{S}\ o \\ \mathsf{npar} & o \ ::= \ \mathtt{S}\ e \end{array}$$

Tym samym par składa się z liczb parzystych takich jak

$$\mathtt{0}, \mathtt{S}\ \mathtt{S}\ \mathtt{0}, \mathtt{S}\ \mathtt{S}\ \mathtt{S}\ \mathtt{S}\ \mathtt{0}, \ldots$$

a npar z liczb nieparzystych takich jak

$$\mathtt{S}\ \mathtt{0}, \mathtt{S}\ \mathtt{S}\ \mathtt{S}\ \mathtt{0}, \mathtt{S}\ \mathtt{S}\ \mathtt{S}\ \mathtt{S}\ \mathtt{S}\ \mathtt{0}, \ldots$$

Zauważmy też , że par oraz npar są **podkategoriami** kategorii nat ponieważ każda liczba parzysta e i każda liczba nieprarzysta o są również liczbami naturalnymi. Tym samym możemy myśleć o par lub npar jako o nat spełniających pewne dodatkowe warunki.

Zadanie 1. *Zdefiniować* par *oraz* npar *niezależnie od siebie.*

Przyjrzyjmy się jeszcze jednemu przykładowi defniowania podkategorii syntaktycznej. Zdefiniujmy najpierw syntaktyczną kategorię paren ciągów nawiasów:

$$\mathsf{paren} \qquad s \ ::= \ \epsilon \mid (s \mid)s$$

ϵ będzie oznaczać pusty ciąg (a więc dla dowolnego ciągu s zachodzi $\epsilon s = s = s\epsilon$). Zauważmy, że paren definiuje język ciągów nawiasów zapisywanych bez żadnych ograniczeń. Teraz zdefiniumy podkategorię mparen kategorii paren ciągów nawiasów, w których nawiasy "lewe" odpowiadają nawiasom "prawym":

$$\mathsf{mparen} \qquad s \ ::= \ \epsilon \mid (s) \mid s \ s$$

mparen generuje ciągi takie jak

$$\epsilon, (), ()(), (()), (())(), ()()(), \ldots$$

Zauważmy, że w myśl przyjętej definicji mparen, ciąg nawiasów z tej kategorii może nie dać się rozłożyć w jednoznaczny sposób: na przykład ciąg ()()() może być równie dobrze otrzymany z dopisania do ciągu () ciągu ()(), jak i z dopisania do ciągu ()() ciągu (). Nietrudno zauważyć, że dwuznaczność ta jest spowodowana przez trzecią część definicji: dla danego ciągu podciągów nawiasów może istnieć więcej niż jeden sposób na rozdzielenie go na dwa podciągu nawiasów. Możemy wyeliminować tę dwuznaczność definiując kategorię lparen:

$$\mathsf{lparen} \qquad s \ ::= \ \epsilon \mid (s) \ s$$

Pomysł na definicję lparen jest następujący: pierwszy (licząc od lewej strony) nawias pojawiający się w niepustym ciągu nawiasów s jest lewy nawias "(", któremu odpowiada dokładnie jedno wystąpienie prawego nawiasu ")". Na przykład ciąg $s = (())()$ może być zapisany jako $(s_1)s_2$, gdzie $s_1 = ()$ oraz $s_2 = ()$ są ciągami odpowiadających sobie lewych i prawych nawiasów, każdy jednoznacznie wyznaczony przez s. Z drugiej strony ciągi ()) oraz (()() nie są ciągami odpowiadających sobie lewych i prawych nawiasów i nie mogą być zapisane w postaci $(s_1)s_2$, gdzie s_1 i s_2 są ciągami odpowiadających nawiasów.

Definicja indukcyjna jest wygodnym sposobem specyfikowania języka. Nawet składnie syntaktyczne języków "na serio" (na przykład Standard ML) używają de facto takiej samej maszynerii. Okazuje się jednak, że inducyjnie definiowane kategorie syntaktyczne nie są najlepszym narzędziem do badania właściwości języka. Na przykład, w jaki w zasadzie sposób chcielibyśmy zapisać, że n należy do nat jeśli n należy do nat – a tym bardziej w jaki sposób chcielibyśmy coś takiego udowodnić? Podobnie, w jaki sposób chcielibyśmy sprawdzić, że dany ciąg nawiasów należy do mparen? Celem udzielenia odpowiedzi na takie pytania musimy najpierw poznać formalną definicję **osądu.**

2.2. **Definiowanie osądów przez indukcję.** Jako **osąd** będziemy rozumieli pewien fragment wiedzy, który może dać się dowieść, lub może nie dać się dowieść. Na przyład:

- "$1 - 1$ jest równe 0" jest osądem, który zawsze daje się dowieść,
- "1 jest równe 0" jest osądem, którego nigdy nie daje się dowieść,
- "dzisiaj pada deszcz" jest osądem, który daje się dowieść w zależności od kontekstu `:-)`
- "`S S 0` należy do kategori syntaktycznej nat" jest osądem, który daje się dowieść o ile nat jest kategorią zdefiniowaną tak, jak przed chwilą.

Jak wobec tego dowodzimy dany osąd? Na przykład, na jakiej podstawie twierdzimy, że "$1-1$ jest równe 0" jest osądem, który zawsze daje się dowieść? Implicite chcielibyśmy użyć prostych reguł arytmetycznych, aby udowodnić, iż "$1-1$ jest równe 0", ale, chcąc być precyzyjnym, musimy wszak odnotować, że reguły arytmetyczne nie są nam dane – musimy je wpierw sformułować jako **reguły wnioskowania.**

Każda **reguła wnioskowania** składa się z osądów $J_1, \ldots, J_n$ zwanych **przesłankami** i z osądu J zwanego **wnioskiem** i zapisywana jest w postaci

$$\frac{J_1 \; J_2 \; \cdots \; J_n}{J} \; R$$

Regułę tę, którą nazwaliśmy tu R, odczytujemy w ten sposób, że jeśli zachodzą przesłanki $J_1, \ldots, J_n$ to zachodzi również wniosek J. Jako szczególny przypadek reguła wnioskowania pozbawiona przesłanej będzie nazywana **aksjomatem.** Oto kilka przykładów reguł wnioskowania – aby nie zaciemniać obrazu, zrezygnowaliśmy z wprowadzania nazw poniższych reguł:

$$\frac{m \textit{ jest równe } l \qquad l \textit{ jest równe } n}{m \textit{ jest równe } n} \qquad \frac{m \textit{ jest równe } l}{m{+}1 \textit{ jest równe } l{+}1}$$

$$\frac{}{m \textit{ jest równe } m} \qquad \frac{}{1 \textit{ jest liczbą naturalną}} \qquad \frac{\textit{mój płaszcz jest mokry}}{\textit{pada deszcz}}$$

Koncepcja osądu jest na tyle ogólna, że w zasadzie stosuje się do dowolnego rodzaju wiedzy: do wiedzy o pogodzie, wiedzy o liczbach, wiedzy o programowaniu etc. Zauważmy wszelako, że same osądy nie są wystarczającym środkiem do uzasadniania fragmentów badanej wiedzy – równie ważne są tu reguły wnioskowania służące do dowodzenia lub obalania osądów. Innymi słowy, aby definicja osądu jest kompletna wtedy i tylko wtedy, gdy istnieją reguły wnioskowania pozwalające na jego udowodnienie bądź obalenie. Bez reguł wnioskowania nie sposób nadać osądowi znaczenie. Na przykład, bez reguł wnioskowania w arytmetyce, stwierdzenie "$1-1$ jest równe 0" nie jest niczym więcej jak ciągiem znaków bez znaczenia i nie może być nazwane osądem.

Jak się domyślamy, osądy są narzędziem wystarczająco silnym aby rozstrzygać o przynależności do kategorii syntaktycznej. Jako przykład spójrzmy na indukcyjną definicję katgorii nat jako na zbiór pewnych osądów i reguł wnioskowania. Na początek wprowadzamy osąd n nat:

$$n \; \mathsf{nat} \qquad \Leftrightarrow \qquad n \textit{ jest liczbą naturalną}$$

oraz następujące dwie reguły wnioskowania, nazwane tu jako *Zero* oraz *Succ:*

$$\frac{}{n \; \mathsf{nat}} \; \textit{Zero} \qquad \frac{n \; \mathsf{nat}}{\mathsf{S} \; n \; \mathsf{nat}} \; \textit{Succ}$$

Występująca w regule *Succ* zmienna n jest zwana **metazmienną** i oznacza pewien ciąg złożony z elementów S oraz 0 i tym samym nie jest częścią języka składającego się z S i 0. Innymi słowy, n jest tu metazmienną przyjmującą wartości ze zbioru ciągów złożonych z S i 0 i (przynajmniej przed zastąpieniem n przez pewien konkretny ciąg, na przykład S 0) nie jest testowana na przynależność do nat.

Tym samym osąd n nat jest teraz zdefiniowany indukcyjnie przez dwie reguły wnioskowania. Reguła *Zero* jest aksjomatem i gra tu rolę przypadku bazowego, zaś reguła *Succ* jest przypadkiem indukcyjnym, albowiem jej przesłanka zawiera osąd o mniejszym rozmiarze niż osąd (tego samego rodzaju) występujący we wniosku. Teraz możemy udowodnić, na przykład, że osąd S S 0 nat ma dowód: robimy to wskazując następujące **drzewo wnioskowania,** w którym S S 0 nat jest korzeniem, a 0 nat jedynym liściem:

$$\dfrac{\dfrac{\dfrac{}{0 \; \mathsf{nat}} \; \textit{Zero}}{\mathsf{S} \; 0 \; \mathsf{nat}} \; \textit{Succ}}{\mathsf{S} \; \mathsf{S} \; 0 \; \mathsf{nat}} \; \textit{Succ}$$

W podobny sposób możemy zapisać definicję kategorii syntaktycznej tree przy użyciu osądów i reguł wnioskowania:

$$t \; \mathsf{tree} \qquad \Leftrightarrow \qquad t \textit{ jest drzewem binarnym liczb naturalnych}$$

$$\frac{n\ \mathsf{nat}}{\mathtt{leaf}\ n\ \mathsf{tree}}\ Leaf \qquad \frac{t_1\ \mathsf{tree} \qquad n\ \mathsf{nat} \qquad t_2\ \mathsf{tree}}{\mathtt{node}\ (t_1, n, t_2)\ \mathsf{tree}}\ Node$$

W troszkę bardziej skomplikowany, ale w sumie też podobny sposób możemy zapisać definicję kategorii syntaktycznej $\mathsf{ctree}\langle d\rangle$ pełnych drzew o danej wysokości:

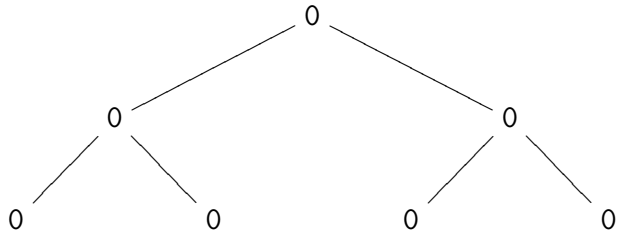
$$t\ \mathsf{ctree}\langle d\rangle \qquad \Leftrightarrow \qquad t \textit{ jest pełnym drzewem binarnym liczb naturalnych o wysokości } \mathrm{d}$$

$$\frac{n\ \mathsf{nat}}{\mathtt{leaf}\ n\ \mathsf{ctree}\langle \mathtt{0}\rangle}\ Cleaf \qquad \frac{t_1\ \mathsf{ctree}\langle d\rangle \qquad n\ \mathsf{nat} \qquad t_2\ \mathsf{ctree}\langle d\rangle}{\mathtt{node}\ (t_1, n, t_2)\ \mathsf{ctree}\langle \mathtt{S}\ d\rangle}\ Cnode$$

Rozważmy następujące drzewo:

$$\mathtt{node}\ (\mathtt{node}\ (\mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{leaf}\ \mathtt{0}),\ \mathtt{0},\ \mathtt{node}\ (\ \mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{leaf}\ \mathtt{0}))$$

i zobaczmy, że jest do istotnie pełne drzewko binarne liczb naturalnych o wysokości $\mathtt{S}\ \mathtt{S}\ \mathtt{0}$. Oczywiście widać to na rysunku:



ale spróbujmy jednak zapisać formalny dowód:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathtt{0}\ \mathsf{nat}}\ Zero}{\mathtt{leaf}\ \mathtt{0}\ \mathsf{ctree}\langle \mathtt{0}\rangle}\ Cleaf \quad \dfrac{}{\mathtt{0}\ \mathsf{nat}}\ Zero \quad \dfrac{\dfrac{}{\mathtt{0}\ \mathsf{nat}}\ Zero}{\mathtt{leaf}\ \mathtt{0}\ \mathsf{ctree}\langle \mathtt{0}\rangle}\ Cleaf}{\mathtt{node}\ (\mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{leaf}\ \mathtt{0})\ \mathsf{ctree}\langle \mathtt{S}\ \mathtt{0}\rangle}\ Cnode \quad \dfrac{}{\mathtt{0}\ \mathsf{nat}}\ Zero \qquad (...)}{\mathtt{node}\ (\mathtt{node}\ (\mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{leaf}\ \mathtt{0}),\ \mathtt{0},\ \mathtt{node}\ (\ \mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{leaf}\ \mathtt{0}))\ \mathsf{ctree}\langle \mathtt{S}\ \mathtt{S}\ \mathtt{0}\rangle}\ Cnode$$

...gdzie w miejsce (...) trzeba wstawić taki sam osąd jak pierwsza z przesłanek w ostatnim wnioskowaniu. Podobnie możemy sprawdzić, że

$$t = \mathtt{node}\ (\mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{node}\ (\ \mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{leaf}\ \mathtt{0}))$$

nie jest pełnym drzewem binarnym, albowiem nie potrafimy udowodnić, iż $t\ \mathsf{ctree}\langle d\rangle$ dla żadnej liczby naturalnej d:

$$\dfrac{\dfrac{\dfrac{}{\mathtt{0}\ \mathsf{nat}}\ Zero \qquad d' = \mathtt{0}}{\mathtt{leaf}\ \mathtt{0}\ \mathsf{ctree}\langle d'\rangle}\ Cleaf \quad \dfrac{}{\mathtt{0}\ \mathsf{nat}}\ Zero \quad \dfrac{\ldots \qquad d' = \mathtt{S}\ d''}{\mathtt{node}\ (\mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{leaf}\ \mathtt{0})\ \mathsf{ctree}\langle d'\rangle}\ Cnode}{\mathtt{node}\ (\mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{node}\ (\ \mathtt{leaf}\ \mathtt{0},\ \mathtt{0},\ \mathtt{leaf}\ \mathtt{0}))\ \mathsf{ctree}\langle \mathtt{S}\ d'\rangle}\ Cnode$$

Łatwo zauważyć, dlaczego dowód nie działa: lewe poddrzewo t wymaga, aby $d' = 0$, tymczasem prawe poddrzewo t wymaga, aby $d' = \mathtt{S}d''$, co oczywiście jest niemożliwe.

Dokładnie tak jak to miało miejsce w przypadku kategorii syntaktycznych par i npar, różne osądy mogą być definiowane równocześnie. Na przykład, definicje par i npar można przedstawić w postaci osądów i reguł wnioskowania w następujący sposób:

$$n\ \mathsf{par} \qquad \Leftrightarrow \qquad n \textit{ jest liczbą parzystą}$$

$$n\ \mathsf{npar} \qquad \Leftrightarrow \qquad n \textit{ jest liczbą nieparzystą}$$

$$\frac{}{\mathtt{0}\ \mathsf{par}}\ ZeroE \qquad \frac{n\ \mathsf{npar}}{\mathtt{S}\ n\ \mathsf{par}}\ SuccE \qquad \frac{n\ \mathsf{par}}{\mathtt{S}\ n\ \mathsf{npar}}\ SuccO$$

W szczególności dowód tego, że S S 0 jest liczbą parzystą wyglądałby następująco:

$$\dfrac{\dfrac{\dfrac{}{\mathsf{0\ par}}\,ZeroE}{\mathsf{S\ 0\ npar}}\,SuccO}{\mathsf{S\ S\ 0\ par}}\,SuccE$$

Zadanie 2. *Zapisać definicje* paren, mparen *i* lparen *przy użyciu osądów i reguł wnioskowania.*

2.3. **Reguły wyprowadzalne i dopuszczalne.** Tak jak przed chwilą zobaczyliśmy, osądy są definiowane wewnątrz pewnego ustalonego systemu reguł wnioskowania. Zauważmy, że dzięki istniejącym regułom wnioskowania możemy tworzyć nowe reguły wnioskowania, które można dodać do systemu. Tak otrzymane nowe reguły nie zmieniają w żaden sposób charakterystyki systemu, jako że wszystkie mogą zostać uzasadnione przy użyciu oryginalnych reguł, ale często mogą pomóc nam w rozwiązywaniu problemów. Na przykład mnożąc dwie liczby naturalne rzadko kiedy używamy podstawowych reguł arytmetyki, o których moglibyśmy myśleć jak o podstawowym zestawie reguł wnioskowania – zamiast tego powszechnie stosujemy tabliczkę mnożenia, o której możemy teraz myśleć jak o zestawie nowych reguł wnioskowania.

Nowe reguły wnioskowania mogą być wprowadzane na dwa sposoby: jako **reguły wyprowadzalne** lub jako **reguły dopuszczalne.** Regułą wyprowadzalną nazywamy taką regułę, w której luka pomiędzy przesłankami a wnioskiem może być wypełniona przez wklejenie określonego drzewa wnioskowania. Innymi słowy, zawsze jesteśmy w stanie wskazać ciąg reguł wnioskowania startujący od wyjściowych przesłanek, a kończący na ostatecznym wniosku. Jako przykład rozważmy następującą regułę orzekającą, że jeśli n jest liczbą naturalna, to jest nią również S S n:

$$\dfrac{n\ \mathsf{nat}}{\mathsf{S\ S}\ n\ \mathsf{nat}}\,Succ2$$

Jest to reguła wyprowadzalna, albowiem możemy ją uzasadnić przy użyciu następującego drzewa dowodowego:

$$\dfrac{\dfrac{n\ \mathsf{nat}}{\mathsf{S}\ n\ \mathsf{nat}}\,Succ}{\mathsf{S\ S}\ n\ \mathsf{nat}}\,Succ$$

Możemy teraz używać reguły $Succ2$ tak, jakby to była oryginalna reguła wnioskowania – w razie czego zawsze możemy podać jej dowód.

Reguła dopuszczalna to taka, w której przesłanka pociąga wniosek. Innymi słowy, jeżeli spełniona jest przesłanka, to spełniony jest też wniosek. Oczywiście każda reguła wyprowadzalna jest dopuszczalna dzięki możliwości formalnego udowodnienia wniosku na podstawie przesłanki. Istnieją wszelako reguły dopuszczalne, które nie są wyprowadzalne. Rozważmy na przykład następującą regułę wnioskowania orzekającą, że jeśli S n jest liczbą naturalną, to jest nią też n:

$$\dfrac{\mathsf{S}\ n\ \mathsf{nat}}{n\ \mathsf{nat}}\,Succ^{-1}$$

Zauważmy po pierwsze, że nie jest to reguła wyprowadzalna: jedyny sposób, aby otrzymać n nat z S n nat to zastosowanie reguły $Succ$, ale przesłanka reguły $Succ$ jest mniejsza niż jej wniosek, podczas gdy S n nat jest większe niż n nat.

Przypuśćmy wszakże, że dysponujemy dowodem przesłanki S n nat. Jako że jedynym sposobem udowodnienia S n nat jest zastosowanie reguły $Succ$, dowód S n nat musi wyglądać jakoś tak:

$$\dfrac{\begin{array}{c}\vdots\\ n\ \mathsf{nat}\end{array}}{\mathsf{S}\ n\ \mathsf{nat}}\,Succ$$

Możemy zatem wyekstrachować mniejsze drzewo dowodowe $\begin{array}{c}\vdots\\ n\ \mathsf{nat}\end{array}$ i tym samym udowodnić n nat. Tym samym pokazaliśmy, że $Succ^{-1}$ jest regułą dopuszczalną.

Istotną cechą reguł wyprowadzalnych jest to, że pozostają one prawdziwe nawet wówczas, gdy wzbogacimy nasz system o nowe reguły. Na przykład reguła $Succ2$ pozostanie prawdziwa niezależnie od tego ile nowych reguł wnioskowania dołożymy, ponieważ dowód osądu S S n nat z osądu n nat jest zawsze możliwy w oparciu o regułę $Succ$.

Nie można, niestety, powiedzieć tego samego o regułach dopuszczalnych. Innymi słowy, reguły dopuszczalne mogą przestać obowiązywać po wzbogaceniu systemu o nowe reguły wnioskowania. Na przykład wyobraźmy sobie, że wyposażymy nasz system w nową regułę

$$\frac{n\ \mathsf{tree}}{\mathsf{S}\ n\ \mathsf{nat}}\ ZKosmosu$$

Reguła $ZKosmosu$ psuje poprzednio dopuszczalną regułę $Succ^{-1}$ ponieważ zastosowanie $Succ$ nie jest teraz jedynym sposobem udowodnienia S n nat i tym samym S n nat nie gwarantuje obecnie n nat. Tym samym prawdziwość każdej z reguł dopuszczalnych musi być weryfikowana za każdym razem, gdy wzbogacamy system o nowe reguły.

Zadanie 3. *Czy reguła* $\frac{n\ \mathsf{par}}{\mathsf{S}\ \mathsf{S}\ n\ \mathsf{par}}\ SuccE^2$ *jest wyprowadzalna, czy dopuszczalna? A reguła* $\frac{\mathsf{S}\ \mathsf{S}\ n\ \mathsf{par}}{n\ \mathsf{par}}\ SuccE^{-2}$*?*