On Ackermann function and inference rules (η) and $(\mathbf{ap_v})$

$$\begin{aligned}\mathrm{Ack}(0,n) &= n+1\\ \mathrm{Ack}(m+1,0) &= \mathrm{Ack}(m,1)\\ \mathrm{Ack}(m+1,n+1) &= \mathrm{Ack}(m,\mathrm{Ack}(m+1,n))\end{aligned}$$

Remark that this is defined by the lexicographic order of (m,n).

1 First term representation: `ack1`

We first give the first representation `ack1` of Ackermann function. In this article, we sometimes write $f(x,y)$ instead of $f(x)(y)$, for readability.

Definition 1.1 (`ack1`) We define `ack1` by the following equation.

$$\mathtt{ack1} = \lambda \mathbf{mn}.\mathrm{cond}(\mathtt{S}\mathbf{n}, \lambda m'.\mathrm{cond}(\mathtt{ack1}(m',1), \lambda n'.\mathtt{ack1}(m',\mathtt{ack1}(\mathtt{S}m',n')))(\mathbf{n}))(\mathbf{m})$$

Checking the behavior of `ack1`

$$\begin{aligned}\mathtt{ack1}(0,n) &\mapsto \mathrm{cond}(\mathtt{S}n, \lambda m'.\mathrm{cond}(\mathtt{ack1}(m',1), \lambda n'.\mathtt{ack1}(m',\mathtt{ack1}(\mathtt{S}m',n')))(n))(0)\\ &\mapsto \mathtt{S}n\end{aligned}$$

$$\begin{aligned}\mathtt{ack1}(\mathtt{S}m,0) &\mapsto \mathrm{cond}(\mathtt{S}0, \lambda m'.\mathrm{cond}(\mathtt{ack1}(m',1), \lambda n'.\mathtt{ack1}(m',\mathtt{ack1}(\mathtt{S}m',n')))(0))(\mathtt{S}m)\\ &\mapsto (\lambda m'.\mathrm{cond}(\mathtt{ack1}(m',1), \lambda n'.\mathtt{ack1}(m',\mathtt{ack1}(\mathtt{S}m',n')))(0))(m)\\ &\mapsto \mathrm{cond}(\mathtt{ack1}(m,1), \lambda n'.\mathtt{ack1}(m',\mathtt{ack1}(\mathtt{S}m',n')))(0)\\ &\mapsto \mathtt{ack1}(m,1)\end{aligned}$$

$$\begin{aligned}\mathtt{ack1}(\underline{\mathtt{S}m},\mathtt{S}n) &\mapsto \mathrm{cond}(\mathtt{SS}n, \lambda m'.\mathrm{cond}(\mathtt{ack1}(m',1), \lambda n'.\mathtt{ack1}(m',\mathtt{ack1}(\mathtt{S}m',n')))(\mathtt{S}n))(\underline{\mathtt{S}m})\\ &\mapsto (\lambda m'.\mathrm{cond}(\mathtt{ack1}(m',1), \lambda n'.\mathtt{ack1}(m',\mathtt{ack1}(\mathtt{S}m',n')))(\mathtt{S}n))(\underline{m})\\ &\mapsto \mathrm{cond}(\mathtt{ack1}(m,1), \lambda n'.\mathtt{ack1}(m,\mathtt{ack1}(\mathtt{S}\underline{m},n')))(\mathtt{S}n)\\ &\mapsto (\lambda n'.\mathtt{ack1}(m,\mathtt{ack1}(\mathtt{S}\underline{m},n')))(n)\\ &\mapsto \mathtt{ack1}(m,\mathtt{ack1}(\mathtt{S}\underline{m},n))\end{aligned}$$

The point of `ack1` is the last case. The term $\underline{\mathtt{S}m}$ of the first line and $\mathtt{S}\underline{m}$ of the last line are slightly diffelent: $\underline{\mathtt{S}m}$ is decomposed into $\underline{m}$ by the cond-reduction, then $\mathtt{S}\underline{m}$ is constructed by substituting m' of $\mathtt{S}m'$ by $\underline{m}$.

1.1 `ack1` $: N \to N \to N$ is not in GTC (with (η))

Claim `ack1` $: N \to N \to N$ is well-typed, but does not satisfy GTC in the system with (η).

$$
\begin{prooftree}
\AxiomC{$n : N \vdash n : N$}
\UnaryInfC{$n : N \vdash \mathtt{S}n : N$}
\AxiomC{$\vdash \mathtt{ack1} : \mathbf{N} \to N \to N$ $(\dagger 1)$}
\RightLabel{(η)}
\UnaryInfC{$m' : \mathbf{N} \vdash \mathtt{ack1}(m') : N \to N$}
\AxiomC{}
\UnaryInfC{$\vdash 1 : N$}
\BinaryInfC{$m' : \mathbf{N} \vdash \mathtt{ack1}(m', 1) : N$}
\AxiomC{$\vdash \mathtt{ack1} : \mathbf{N} \to \mathbf{N} \to N$ $(\dagger 2)$}
\RightLabel{(η)}
\UnaryInfC{$m' : \mathbf{N} \vdash \mathtt{ack1}(m') : \mathbf{N} \to N$}
\AxiomC{$\vdash \mathtt{ack1} : N \to \mathbf{N} \to N$ $(\dagger 3)$}
\AxiomC{}
\UnaryInfC{$m' : \mathbf{N} \vdash m' : N$}
\UnaryInfC{$m' : \mathbf{N} \vdash \mathtt{S}m' : N$}
\RightLabel{$(*)$}
\BinaryInfC{$m' : \mathbf{N} \vdash \mathtt{ack1}(\mathtt{S}m') : \mathbf{N} \to N$}
\RightLabel{(η)}
\UnaryInfC{$m' : \mathbf{N}, n' : \mathbf{N} \vdash \mathtt{ack1}(\mathtt{S}m', n') : N$}
\BinaryInfC{$m' : \mathbf{N}, n' : \mathbf{N} \vdash \mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')) : N$}
\UnaryInfC{$m' : \mathbf{N} \vdash \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')) : \mathbf{N} \to N$}
\RightLabel{(cond)}
\BinaryInfC{$m' : \mathbf{N} \vdash \mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n'))) : \mathbf{N} \to N$}
\RightLabel{(η)}
\UnaryInfC{$n : \mathbf{N}, m' : \mathbf{N} \vdash \mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')))(n) : N$}
\UnaryInfC{$n : \mathbf{N} \vdash \lambda m'.\mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')))(n) : \mathbf{N} \to N$}
\RightLabel{(cond)}
\BinaryInfC{$n : \mathbf{N} \vdash \mathrm{cond}(\mathtt{S}n, \lambda m'.\mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')))(n)) : \mathbf{N} \to N$}
\RightLabel{(η)}
\UnaryInfC{$m : \mathbf{N}, n : \mathbf{N} \vdash \mathrm{cond}(\mathtt{S}n, \lambda m'.\mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')))(n))(m) : N$}
\UnaryInfC{$\vdash \mathtt{ack1} : \mathbf{N} \to \mathbf{N} \to N$ $(\dagger)$}
\end{prooftree}
$$

The problem is (ap). This cuts the trace chasing of $\mathbf{N}$. The infinite path $(\dagger) \rightsquigarrow (\dagger 1) \rightsquigarrow (\dagger) \rightsquigarrow (\dagger 3) \rightsquigarrow (\dagger) \rightsquigarrow (\dagger 1) \rightsquigarrow (\dagger) \rightsquigarrow (\dagger 3) \rightsquigarrow \cdots$ does not contains progressing trace.

1.2 $\mathtt{ack1} : N \to N \to N$ is not in GTC (with $(\mathrm{ap_v})$)

Claim $\mathtt{ack1} : N \to N \to N$ is well-typed, but does not satisfy GTC in the system with $(\mathrm{ap_v})$.

The situation is the same as before. In the following, weakening is implicitly applied.

$$
\begin{prooftree}
\AxiomC{$n : N \vdash n : N$}
\UnaryInfC{$n : N \vdash \mathtt{S}n : N$}
\AxiomC{$\vdash \mathtt{ack1} : \mathbf{N} \to N \to N$ $(\dagger 1)$}
\UnaryInfC{$m' : N \vdash \mathtt{ack1} : \mathbf{N} \to N \to N$}
\RightLabel{$(\mathrm{ap_v})$}
\UnaryInfC{$m' : \mathbf{N} \vdash \mathtt{ack1}(m') : N \to N$}
\AxiomC{}
\UnaryInfC{$\vdash 1 : N$}
\BinaryInfC{$m' : \mathbf{N} \vdash \mathtt{ack1}(m', 1) : N$}
\AxiomC{$\vdash \mathtt{ack1} : \mathbf{N} \to \mathbf{N} \to N$ $(\dagger 2)$}
\UnaryInfC{$m' : N \vdash \mathtt{ack1} : \mathbf{N} \to \mathbf{N} \to N$}
\RightLabel{$(\mathrm{ap_v})$}
\UnaryInfC{$m' : \mathbf{N} \vdash \mathtt{ack1}(m') : \mathbf{N} \to N$}
\AxiomC{$\vdash \mathtt{ack1} : N \to \mathbf{N} \to N$ $(\dagger 3)$}
\AxiomC{}
\UnaryInfC{$m' : \mathbf{N} \vdash m' : N$}
\UnaryInfC{$m' : \mathbf{N} \vdash \mathtt{S}m' : N$}
\RightLabel{$(\mathrm{ap_{\neg v}})$}
\BinaryInfC{$m' : \mathbf{N}, n' : N \vdash \mathtt{ack1}(\mathtt{S}m') : \mathbf{N} \to N$}
\RightLabel{$(\mathrm{ap_v})$}
\UnaryInfC{$m' : \mathbf{N}, n' : \mathbf{N} \vdash \mathtt{ack1}(\mathtt{S}m', n') : N$}
\BinaryInfC{$m' : \mathbf{N}, n' : \mathbf{N} \vdash \mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')) : N$}
\UnaryInfC{$m' : \mathbf{N} \vdash \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')) : \mathbf{N} \to N$}
\RightLabel{(cond)}
\BinaryInfC{$n : N, m' : \mathbf{N} \vdash \mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n'))) : \mathbf{N} \to N$}
\RightLabel{$(\mathrm{ap_v})$}
\UnaryInfC{$n : \mathbf{N}, m' : \mathbf{N} \vdash \mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')))(n) : N$}
\UnaryInfC{$n : \mathbf{N} \vdash \lambda m'.\mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')))(n) : \mathbf{N} \to N$}
\RightLabel{(cond)}
\BinaryInfC{$m : N, n : \mathbf{N} \vdash \mathrm{cond}(\mathtt{S}n, \lambda m'.\mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')))(n)) : \mathbf{N} \to N$}
\RightLabel{$(\mathrm{ap_v})$}
\UnaryInfC{$m : \mathbf{N}, n : \mathbf{N} \vdash \mathrm{cond}(\mathtt{S}n, \lambda m'.\mathrm{cond}(\mathtt{ack1}(m', 1), \lambda n'.\mathtt{ack1}(m', \mathtt{ack1}(\mathtt{S}m', n')))(n))(m) : N$}
\UnaryInfC{$\vdash \mathtt{ack1} : \mathbf{N} \to \mathbf{N} \to N$ $(\dagger)$}
\end{prooftree}
$$

2 Second term representation: $\mathtt{ack2}$

We give the second representation $\mathtt{ack2}$ of Ackermann function.

Definition 2.1 ($\mathtt{ack2}$) We define $\mathtt{ack2}$ by the following equation.

$$\mathtt{ack2} = \lambda \mathbf{mn}.\mathrm{cond}(\mathtt{S}\mathbf{n}, \lambda m'.\mathrm{cond}(\mathtt{ack2}(m', 1), \lambda n'.\mathtt{ack2}(m', \mathtt{ack2}(\mathbf{m}, n')))(\mathbf{n}))(\mathbf{m})$$

Checking the behavior of `ack2`

We check only the last case.

$$
\begin{aligned}
\texttt{ack2}(\underline{\texttt{S}m}, \texttt{S}n) &\mapsto \mathrm{cond}(\texttt{SS}n, \lambda m'.\mathrm{cond}(\texttt{ack2}(m', 1), \lambda n'.\texttt{ack2}(m', \texttt{ack2}(\underline{\texttt{S}m}, n')))(\texttt{S}n))(\underline{\texttt{S}m}) \\
&\mapsto (\lambda m'.\mathrm{cond}(\texttt{ack2}(m', 1), \lambda n'.\texttt{ack2}(m', \texttt{ack2}(\underline{\texttt{S}m}, n')))(\texttt{S}n))(\underline{m}) \\
&\mapsto \mathrm{cond}(\texttt{ack2}(m, 1), \lambda n'.\texttt{ack2}(m, \texttt{ack2}(\underline{\texttt{S}m}, n')))(\texttt{S}n) \\
&\mapsto (\lambda n'.\texttt{ack2}(m, \texttt{ack2}(\underline{\texttt{S}m}, n')))(n) \\
&\mapsto \texttt{ack2}(m, \texttt{ack2}(\underline{\texttt{S}m}, n))
\end{aligned}
$$

The point of `ack2` is that $\underline{\texttt{S}m}$ at the first line and the one at the last line are exactly the same.

2.1 $\texttt{ack2} : N \to N \to N$ is not in GTC (with (η))

Claim $\texttt{ack2} : N \to N \to N$ is well-typed, but does not satisfy GTC in the system with (η).

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$n : N \vdash n : N$}
\UnaryInfC{$n : N \vdash \texttt{S}n : N$}
\AxiomC{$(\dagger 1)$}
\noLine
\UnaryInfC{$\vdash \texttt{ack2} : N \to N \to N$}
\RightLabel{(η)}
\UnaryInfC{$m' : N \vdash \texttt{ack2}(m') : N \to N$}
\AxiomC{}
\UnaryInfC{$\vdash 1 : N$}
\BinaryInfC{$m' : N \vdash \texttt{ack2}(m', 1) : N$}
\AxiomC{$(\dagger 2)$}
\noLine
\UnaryInfC{$\vdash \texttt{ack2} : N \to N \to N$}
\RightLabel{(η)}
\UnaryInfC{$m' : N \vdash \texttt{ack2}(m') : N \to N$}
\AxiomC{$(\dagger 3)$}
\noLine
\UnaryInfC{$\vdash \texttt{ack2} : {\color{red}\mathbf{N}} \to {\color{olive}\mathbf{N}} \to N$}
\RightLabel{(η)}
\UnaryInfC{$m : {\color{red}\mathbf{N}} \vdash \texttt{ack2}(m) : {\color{olive}\mathbf{N}} \to N$}
\RightLabel{(η)}
\UnaryInfC{$m : {\color{red}\mathbf{N}}, n' : {\color{olive}\mathbf{N}} \vdash \texttt{ack2}(m, n') : N$}
\BinaryInfC{$m : {\color{red}\mathbf{N}}, m' : N, n' : {\color{olive}\mathbf{N}} \vdash \texttt{ack2}(m', \texttt{ack2}(m, n')) : N$}
\UnaryInfC{$m : {\color{red}\mathbf{N}}, m' : N \vdash \lambda n'.\texttt{ack2}(m', \texttt{ack2}(m, n')) : {\color{olive}\mathbf{N}} \to N$}
\RightLabel{(cond)}
\BinaryInfC{$m : {\color{red}\mathbf{N}}, m' : N \vdash \mathrm{cond}(\texttt{ack2}(m', 1), \lambda n'.\texttt{ack2}(m', \texttt{ack2}(m, n'))) : {\color{olive}\mathbf{N}} \to N$}
\RightLabel{(η)}
\UnaryInfC{$m : {\color{red}\mathbf{N}}, n : {\color{olive}\mathbf{N}}, m' : N \vdash \mathrm{cond}(\texttt{ack2}(m', 1), \lambda n'.\texttt{ack2}(m', \texttt{ack2}(m, n')))(n) : N$}
\UnaryInfC{$m : {\color{red}\mathbf{N}}, n : {\color{olive}\mathbf{N}} \vdash \lambda m'.\mathrm{cond}(\texttt{ack2}(m', 1), \lambda n'.\texttt{ack2}(m', \texttt{ack2}(m, n')))(n) : N \to N$}
\RightLabel{(cond)}
\BinaryInfC{$m : {\color{red}\mathbf{N}}, n : {\color{olive}\mathbf{N}} \vdash \mathrm{cond}(\texttt{S}n, \lambda m'.\mathrm{cond}(\texttt{ack2}(m', 1), \lambda n'.\texttt{ack2}(m', \texttt{ack2}(m, n')))(n)) : N \to N$}
\AxiomC{}
\UnaryInfC{$m : {\color{red}\mathbf{N}} \vdash m : N$}
\RightLabel{(ap)}
\BinaryInfC{$m : {\color{red}\mathbf{N}}, n : {\color{olive}\mathbf{N}} \vdash \mathrm{cond}(\texttt{S}n, \lambda m'.\mathrm{cond}(\texttt{ack2}(m', 1), \lambda n'.\texttt{ack2}(m', \texttt{ack2}(m, n')))(n))(m) : N$}
\UnaryInfC{$\vdash \texttt{ack2} : {\color{red}\mathbf{N}} \to {\color{olive}\mathbf{N}} \to N \ (\dagger)$}
\end{prooftree}
$$

The problem is (ap). This cuts the trace chasing of ${\color{red}\mathbf{N}}$. The infinite path $(\dagger) \rightsquigarrow (\dagger 1) \rightsquigarrow (\dagger) \rightsquigarrow (\dagger 1) \rightsquigarrow \cdots$ does not contains progressing trace.

2.2 $\texttt{ack2} : N \to N \to N$ is not in GTC (with $(\mathrm{ap_v})$)

Claim $\texttt{ack2} : N \to N \to N$ is well-typed, but does not satisfy GTC in the system with $(\mathrm{ap_v})$.

$$
\dfrac{
\dfrac{\dfrac{}{n : N \vdash n : N}}{n : N \vdash \mathtt{S}n : N}
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\overset{(\dagger 1)}{\vdash \mathtt{ack2} : {\color{red}\mathbf{N}} \to N \to N}}{m' : {\color{red}\mathbf{N}} \vdash \mathtt{ack2}(m') : N \to N}\;(\mathrm{ap_v})
\quad
\dfrac{}{\vdash 1 : N}
}{m' : {\color{red}\mathbf{N}} \vdash \mathtt{ack2}(m', 1) : N}
\quad
\dfrac{
\dfrac{
\dfrac{\overset{(\dagger 2)}{\vdash \mathtt{ack2} : {\color{red}\mathbf{N}} \to N \to N}}{m' : {\color{red}\mathbf{N}} \vdash \mathtt{ack2}(m') : N \to N}\;(\eta)
\quad
\dfrac{\dfrac{\overset{(\dagger 3)}{\vdash \mathtt{ack2} : N \to {\color{olive}\mathbf{N}} \to N}}{m : N \vdash \mathtt{ack2}(m) : {\color{olive}\mathbf{N}} \to N}\;(\eta)}{m : N, n' : {\color{olive}\mathbf{N}} \vdash \mathtt{ack2}(m, n') : N}\;(\mathrm{ap_v})
}{m : N, m' : {\color{red}\mathbf{N}}, n' : {\color{olive}\mathbf{N}} \vdash \mathtt{ack2}(m', \mathtt{ack2}(m, n')) : N}
}{m : N, m' : {\color{red}\mathbf{N}} \vdash \lambda n'.\mathtt{ack2}(m', \mathtt{ack2}(m, n')) : {\color{olive}\mathbf{N}} \to N}
}{m : N, n : N, m' : {\color{red}\mathbf{N}} \vdash \mathrm{cond}(\mathtt{ack2}(m', 1), \lambda n'.\mathtt{ack2}(m', \mathtt{ack2}(m, n'))) : {\color{olive}\mathbf{N}} \to N}\;(\mathrm{cond})
}{m : N, n : {\color{olive}\mathbf{N}}, m' : {\color{red}\mathbf{N}} \vdash \mathrm{cond}(\mathtt{ack2}(m', 1), \lambda n'.\mathtt{ack2}(m', \mathtt{ack2}(m, n')))(n) : N}\;(\mathrm{ap_v})
}{m : N, n : {\color{olive}\mathbf{N}} \vdash \lambda m'.\mathrm{cond}(\mathtt{ack2}(m', 1), \lambda n'.\mathtt{ack2}(m', \mathtt{ack2}(m, n')))(n) : {\color{red}\mathbf{N}} \to N}
}{
\dfrac{
\dfrac{m : N, n : {\color{olive}\mathbf{N}} \vdash \mathrm{cond}(\mathtt{S}n, \lambda m'.\mathrm{cond}(\mathtt{ack2}(m', 1), \lambda n'.\mathtt{ack2}(m', \mathtt{ack2}(m, n')))(n)) : {\color{red}\mathbf{N}} \to N}{m : {\color{red}\mathbf{N}}, n : {\color{olive}\mathbf{N}} \vdash \mathrm{cond}(\mathtt{S}n, \lambda m'.\mathrm{cond}(\mathtt{ack2}(m', 1), \lambda n'.\mathtt{ack2}(m', \mathtt{ack2}(m, n')))(n))(m) : N}\;(\mathrm{ap_v})
}{\vdash \mathtt{ack2} : {\color{red}\mathbf{N}} \to {\color{olive}\mathbf{N}} \to N \;(\dagger)}
}\;(\mathrm{cond})
$$

The problem is the lower most $(\mathrm{ap_v})$. This cuts the trace chasing of $m : {\color{red}\mathbf{N}}$. Perhaps $m : N$ at $(\dagger 3)$ should be ${\color{red}\mathbf{N}}$.

Summing up, both (η) and $(\mathrm{ap_v})$ are problematic. Possible solution might be:

$$
\dfrac{\Gamma, x : {\color{red}\mathbf{N}} \vdash f : {\color{red}\mathbf{N}} \to A}{\Gamma, x : {\color{red}\mathbf{N}} \vdash f(x) : A}\;(\mathrm{ap_v})
$$