CSI 5109 Assignment 4

1. By constructing the refusal trees of the two behaviour expressions below, show that conformance is not a symmetric relation.

\[ A = a; b; \text{stop} \] \[ B = i; a; \text{stop} \] \[ b; c; \text{stop} \]

\[
\text{Tr}(A) = \{ \epsilon, a, ab, c\}
\]
\[
\text{Tr}(B) = \{ \epsilon, a, b, bc\}
\]

\[
\text{Tr}(A) \cap \text{Tr}(B) = \{ \epsilon, a\}
\]

\[
\text{Ref}(A, \epsilon) = \{ b\}
\]
\[
\text{Ref}(B, \epsilon) = \{ b, c\}
\]

\[
\text{Ref}(A, \epsilon) \subseteq \text{Ref}(B, \epsilon)
\]

\[
\text{Ref}(A, a) = \{ a, c\}
\]
\[
\text{Ref}(B, a) = \{ a, b, c\}
\]

\[
\text{Ref}(A, a) \subseteq \text{Ref}(B, a)
\]

A conf B since \( \text{Ref}(A, \epsilon) \subseteq \text{Ref}(B, \epsilon) \) and \( \text{Ref}(A, a) \subseteq \text{Ref}(B, a) \); but B does not conform to A since \( \text{Ref}(B, \epsilon) \nsubseteq \text{Ref}(A, \epsilon) \) (not to mention that \( \text{Ref}(B, a) \nsubseteq \text{Ref}(A, a) \) either).

\Rightarrow \text{CONF is NOT a symmetric relation.}
2. Given the behaviour expressions:
   \[ A = (a; (b; \textit{stop} [i; c; \textit{stop})] [a,c]; (a; (i; b; \textit{stop} [i; c; \textit{stop})) \]
   \[ B = (a; (b; \textit{stop} [c; \textit{stop})] [a, c]; (a; (i; b; \textit{stop} [i; c; \textit{stop})) \]

   a) Are \( A \) and \( B \) weak bisimulation equivalent?

   If \( A \) and \( B \) are derived on \( a \):
   \[ A \rightarrow_{a} A' = (b; \textit{stop} [i; c; \textit{stop})] [a,c]; (i; b; \textit{stop} [i; c; \textit{stop}) \]
   \[ B \rightarrow_{a} B' = (b; \textit{stop} [c; \textit{stop})] [a, c]; (i; b; \textit{stop} [i; c; \textit{stop}) \]

   Then \( A' \) (executing \( i \) on left side) and \( B' \) (executing first \( i \) on right side) on \( \varepsilon \):
   \[ A' \rightarrow_{\varepsilon} A'' = (c; \textit{stop}) [a,c]; (i; b; \textit{stop} [i; c; \textit{stop}) \]
   \[ B' \rightarrow_{\varepsilon} B'' = (b; \textit{stop} [c; \textit{stop})] [a, c]; (b; \textit{stop}) \]

   The resulting \( A'' \) can execute \( c \) but \( B'' \) cannot \( \Rightarrow \) \( A \) is NOT weak bisimilar to \( B \).

   b) Does one of them conform to the other?

   \[ \text{Ref}(A) = \text{Ref}(B) = \{a, b, c\} \]

   \[ \text{Tr}(A) = \text{Tr}(B) = \text{Tr}(A) \cap \text{Tr}(B) = \{\varepsilon, a, ab, abb, ac\} \]

   \[ \text{Ref}(A,\varepsilon) = \text{Ref}(B,\varepsilon) = \{b, c\} \Rightarrow \text{Ref}(A,\varepsilon) \subseteq \text{Ref}(B,\varepsilon) \text{ and } \text{Ref}(B,\varepsilon) \subseteq \text{Ref}(A,\varepsilon) \]

   \[ \text{Ref}(A,a) = \text{Ref}(B,a) = \{a, c\} \Rightarrow \text{Ref}(A,a) \subseteq \text{Ref}(B,a) \text{ and } \text{Ref}(B,a) \subseteq \text{Ref}(A,a) \]

   \[ \text{Ref}(A,ab) = \text{Ref}(B,ab) = \{a, b, c\} \Rightarrow \text{Ref}(A,ab) \subseteq \text{Ref}(B,ab) \text{ and } \text{Ref}(B,ab) \subseteq \text{Ref}(A,ab) \]

   \[ \text{Ref}(A,ac) = \text{Ref}(B,ac) = \{a, b, c\} \Rightarrow \text{Ref}(A,ac) \subseteq \text{Ref}(B,ac) \text{ and } \text{Ref}(B,ac) \subseteq \text{Ref}(A,ac) \]

   Thus, \( A \) conf \( B \) and \( B \) conf \( A \).

   c) Are they trace equivalent?

   Yes. As shown above: \( \text{Tr}(A) = \text{Tr}(B) = \{\varepsilon, a, ab, abb, ac\} \)

   d) Are they testing equivalent?

   Since \( A \) conf \( B \) and \( B \) conf \( A \) and they share the same traces, \( A \) te \( B \).
3. Construct the canonical tester of the following behaviour expression and derive the set of test cases:

\[(a; (b; \text{stop} [] c; \text{stop})) || [a,c] || (a; (i; b; \text{stop} [] c; \text{stop}))\]

\[L = \{a,b,c\}\]
\[L^* = \{\{\},\{a\},\{b\},\{c\},\{a,b\},\{a,c\},\{b,c\},\{a,b,c\}\}\]

\[\text{Tr}(S) = \{\varepsilon, a, ab, abb, ac\}\]

\[\begin{align*}
\text{Ref}(S, \varepsilon) &= \{\{\},\{a\},\{b\},\{c\}\} \\
\text{Ref}(S, a) &= \{\{\},\{a\},\{c\}\} \\
\text{Ref}(S, ab) &= \{\{\},\{a\},\{c\},\{a,c\}\} \\
\text{Ref}(S, abb) &= L^* \\
\text{Ref}(T(S), \varepsilon) &= \text{Ref}(S, \varepsilon) \\
\text{Ref}(T(S), a) &= \text{Ref}(S, a) \\
\text{Ref}(T(S), ab) &= \text{Ref}(S, ab) \\
\text{Ref}(T(S), abb) &= L^* \\
\text{Ref}(T(S), ac) &= L^*
\end{align*}\]

where \(\forall \sigma \in \text{Ref}(T(S), \varepsilon), \{a,b,c\} \in \text{Ref}(S, \varepsilon) \iff \{a,b,c\}\sigma \in \text{Ref}(S, \varepsilon)\)

where \(\forall \sigma \in \text{Ref}(T(S), a), \{a,b,c\} \in \text{Ref}(S, a) \iff \{a,b,c\}\sigma \in \text{Ref}(S, a)\)

where \(\forall \sigma \in \text{Ref}(T(S), ab), \{a,b,c\} \in \text{Ref}(S, ab) \iff \{a,b,c\}\sigma \in \text{Ref}(S, ab)\)

where \(\forall \sigma \in \text{Ref}(T(S), abb), \{a,b,c\} \in \text{Ref}(S, abb) \iff \{a,b,c\}\sigma \in \text{Ref}(S, abb)\)

where \(\forall \sigma \in \text{Ref}(T(S), ac), \{a,b,c\} \in \text{Ref}(S, ac) \iff \{a,b,c\}\sigma \in \text{Ref}(S, ac)\)
4. By reference to the notes by Burstall, prove the following:

a) For all m, n, \( m + \text{succ}(n) = \text{succ}(m+n) \)  \[\text{Proposition 5.2 on page 7}\]

lemmas:
\[
\begin{align*}
0 + n &= n & (1) \\
\text{succ}(m) + n &= \text{succ}(m + n) & (2)
\end{align*}
\]

I.H.
\[
m + \text{succ}(n) = \text{succ}(m + n)
\]

Base: \( m = 0 \)
\[
0 + \text{succ}(n) = \text{succ}(0 + n)
\]
LHS reduced by lemma (1):
\[
\text{succ}(n) = \text{succ}(0 + n)
\]
RHS reduced by lemma (1):
\[
\text{succ}(n) = \text{succ}(n)
\]

Step: \( m \rightarrow \text{succ}(m) \)
\[
\text{succ}(m) + \text{succ}(n) = \text{succ}(\text{succ}(m) + n)
\]
LHS by lemma (2):
\[
\text{succ}(m + \text{succ}(n)) = \text{succ}(\text{succ}(m) + n)
\]
LHS by I.H.:
\[
\text{succ}(\text{succ}(m + n)) = \text{succ}(\text{succ}(m) + n)
\]
RHS by lemma (2):
\[
\text{succ}(\text{succ}(m + n)) = \text{succ}(\text{succ}(m + n))
\]

b) For all l, \( \text{join}(l,\text{nil}) = l \)  \[\text{first Lemma 6.1 on page 9}\]

lemmas:
\[
\begin{align*}
\text{join}(\text{nil},l) &= l & (1) \\
\text{join}(s::k,l) &= s::\text{join}(k,l) & (2)
\end{align*}
\]

I.H.:
\[
\text{join}(l,\text{nil}) = l
\]

Base: \( l = \text{nil} \)
\[
\text{join}(\text{nil},\text{nil}) = \text{nil}
\]
LHS by lemma (1):
\[
\text{nil} = \text{nil}
\]

Step: \( l \rightarrow s::l \)
\[
\text{join}(s::l,\text{nil}) = s::l
\]
LHS by lemma (2):
\[
s::\text{join}(l,\text{nil}) = s::l
\]
LHS by I.H.:
\[
s::l = s::l
\]
c) Given the definition:
   \[
   \text{length : list(alpha) -> nat}
   \]
   \[
   \text{length(nil) <= 0} \quad (1)
   \]
   \[
   \text{length(n::l) <= length(l) + 1} \quad (2)
   \]

Prove that:
\[
\text{length(join(k,l)) = length(k) + length(l)}
\]

lemmas:
\[
0 + n = n \quad (3)
\]
\[
\text{join(nil,l) = l} \quad (4)
\]
\[
\text{join(s::k,l) = s::join(k,l)} \quad (5)
\]
\[
\forall m,n \in \text{nat}, m + n = n + m \quad (6)
\]

I.H.
\[
\text{length(join(k,l)) = length(k) + length(l)}
\]

Base: \( k = \text{nil} \)
\[
\text{length(join(nil,l)) = length(nil) + length(l)}
\]
RHS by (1):
\[
\text{length(join(nil,l)) = 0 + length(l)}
\]
RHS by lemma (3):
\[
\text{length(join(nil,l)) = length(l)}
\]
LHS by lemma (4):
\[
\text{length(l) = length(l)}
\]

Step: \( k -> s::k \)
\[
\text{length(join(s::k,l)) = length(s::k) + length(l)}
\]
RHS by (2):
\[
\text{length(join(s::k,l)) = length(k) + 1 + length(l)}
\]
LHS by lemma (5):
\[
\text{length(s::join(k,l)) = length(k) + 1 + length(l)}
\]
LHS by (2):
\[
\text{length(join(k,l)) + 1 = length(k) + 1 + length(l)}
\]
LHS by I.H.:
\[
\text{length(k) + length(l) + 1 = length(k) + 1 + length(l)}
\]
LHS by lemma (6):
\[
\text{length(k) + 1 + length(l) = length(k) + 1 + length(l)}
\]