Theorem: If every path of \( N \) contains a step needed at \( M_0 \), then \( (N, M_0) \) is live.

Proof: Let \( M \) be a marking of \( N \). A transient is:
- dead at \( M \) if it is not enabled at any marking reachable from \( M \).
- live at \( M \) if it is "cannot die", i.e., it is not (invariantly) dead at any marking reachable from \( M \).
- marked at \( M \) if there is \( M \rightarrow M' \) such that it is dead at \( M' \).

Claim: There is a reachable marking \( M' \) such that every transient is either dead or marked at \( M' \).

Claim: For every \( t \) dead at \( M \), there exists some \( s \in T \) such that:
- \( M(s) = 0 \) and
- every \( t' \in s' \) is dead at \( M \).

Proof of the claim: Let \( t \) be dead at \( M \) and let \( \{s_1, \ldots, s_n\} \) be the places in \( t \) not marked at \( M \). Assume that for every \( s_i \), there is \( t_i \in s_i \) such that \( M \rightarrow M_i \). Since \( N \) is free-choice, there are markings \( M_i \rightarrow M_i \) such that:

Conclude why that \( t \) is dead at \( M' \).
Theorem. If $(M, M_0)$ is free, dense and live, then every region of $M$ contains a trap located at $M_0$.

Proof sketch. Let $R$ be a region, and let $Q$ be the maximal trap included in $R$. Let $D = R \setminus Q$.

Assume $Q$ is actually connected.

We find a strongly connected $\tau$ that empties $D$ without adding states to $Q$. After the execution of $\tau$, $R$ is empty, which implies that $(M, M_0)$ is not live.

The sequence $\tau$ is constructed as follows:

- We construct an allocation that ensures to each place $S$ a transition of $S$. The firing sequence $\tau$ only contains allocated transitions.

We need to guarantee:

- the allocation does not define cycles
- the allocation does not allocate any transition of $Q$ (this would undo the trap)
- while there are tokens in $D$ we can always fire allocated transitions you and you
Definition: Let $x$ be a place or transition of a net $N = (S, T, F)$. The cluster of $x$, denoted by $[x]$, is the minimal set of nodes such that:
- $x \in [x]$
- If $s \in [x] \cap S$ then $s \subseteq [x]$
- If $t \in [x] \cap T$ then $t \subseteq [x]$

Proposition: Every node of a net belongs to exactly one cluster.

The set of clusters is a partition of $S \cup T$.

Definition: Let $N = (S, T, F)$ be a net, and let $C$ be a set of clusters of $N$. An allocation of $C$ is a function $\alpha : C \rightarrow T$ satisfying $\alpha(c) \in c$.

Proposition: Let $N$ be a place-choice net, let $R$ be a set of places, let $Q$ be the maximal trap included in $R$, and let $D = R \setminus Q$.

Let $C = \{ [t] \mid t \in D^+ \}$.

There exists a circuit-free allocation $\alpha : C \rightarrow T$ such that $\alpha(c) \subseteq Q$.

(circuit-free: the set of runs $\{ (s, \nu(o)) \mid s \in D \} \cup F \cap (T \times \bar{I})$ does not contain any circuit)

Proof:

By induction on $|R|$.

If $|R| = 0$, then $C = \emptyset$.

If $|R| > 0$, if $R$ is a trap then $C = \emptyset$.

If $R$ is not a trap, then:

Let $R^1 = R \setminus \{t\}$, $D^1 = R^1 \setminus Q^1$, $Q^1 = \{ [t] \mid t \in D^+ \}$.

By induction hypothesis, there exists $\alpha^1 : C^1 \rightarrow T$ circuit-free for $D^1$ such that $\alpha^1(c) \subseteq D^1$.

Define $\alpha : C \rightarrow T$ by

$$\alpha(c) = \begin{cases} \alpha^1(c) & \text{if } c \neq [t] \\ \{t\} & \text{if } c = [t] \end{cases}$$

We have to prove:
- $\alpha(C) = \alpha^1(C^1) \cup \{t\}$ (well-defined)
- $\alpha$ is circuit-free for $D$
- $\alpha$ puts $t$ in $Q$ whenever $t \in Q$. 
Proposition: Let \( \pi \) be an allocation of a logic-free choice system with domain \( C \). Then \( \pi \) is in infinite occurrence sequence \( S \) if and only if

- \( \pi \) never allocated transition infinitely often
- \( \pi \) never fires any non-allocated transition of \( C \)

Proof: Immediate consequence of

\[
\begin{array}{c}
\text{Condition: The Hansen pattern for free-choice net}
\end{array}
\]

is \( \text{NP} \)-complete

Proof: \( \text{NP} \)-hardness: by reduction from \( \text{SAT} \)

- Membership in \( \text{NP} \): given a system (or just its \( \pi \)
- Compute the legal tree on the \( \pi \)
- Check: the tree is empty at \( \mu_0 \)