Petri nets

- Introduced in 1962 (though claimed to have been invented by 1939)
- Starting point: think of a transition system where a number of processes can be in a given state and then allow coordination
- **Conditions**: local components of state
- **Events**: transitions and coordination
- Allows study of concurrency of events, reasoning about causal dependency and how the action of one process might conflict with that of another
- The first of a range of models: event structures, Mazurkiewicz trace languages, asynchronous transition systems, ... 
- Many variants with different algorithmic properties and expressivity
Multisets generalise sets by allow elements to occur some number of times. \(\infty\)-multisets generalise further by allowing infinitely many occurrences.

\[
\omega^\infty = \omega \cup \{\infty\}
\]

Extend addition:
\[
n + \infty = \infty \quad \text{for } n \in \omega^\infty
\]

Extend subtraction
\[
\infty - n = \infty \quad \text{for } n \in \omega
\]

Extend order:
\[
n \leq \infty \quad \text{for } n \in \omega^\infty
\]

An \(\infty\)-multiset over a set \(X\) is a function

\[
f : X \to \omega^\infty
\]

It is a multiset if \(f : X \to \omega\).
Operations on $\infty$-multisets

- $f \leq g$ iff $\forall x \in X. f(x) \leq g(x)$
- $f + g$ is the $\infty$-multiset such that
  \[ \forall x \in X. (f + g)(x) = f(x) + g(x) \]
- For $g$ a multiset such that $g \leq f$,
  \[ \forall x \in X. (f - g)(x) = f(x) - g(x) \]
General Petri nets

A general Petri net consists of

- a set of conditions $P$
- a set of events $T$
- a pre-condition map assigning to each event $t$ a multiset of conditions $\bullet t$
- a post-condition map assigning to each event $t$ an $\infty$-multiset of conditions $t^*$
- a capacity map $Cap$ an $\infty$-multiset of conditions, assigning a capacity in $\omega^\infty$ to each condition
A marking is an $\infty$-multiset $M$ such that

$$M \leq Cap$$

giving how many tokens are in each condition.

The token game:

For $M, M'$ markings, $t$ an event:

$$M \xrightarrow{t} M' \quad \text{iff} \quad \bullet t \leq M \quad \& \quad M' = M - \bullet t + t\bullet$$

An event $t$ has concession (is enabled) at $M$ iff

$$\bullet t \leq M \quad \& \quad M - \bullet t + t\bullet \leq Cap$$
Further examples
Often don’t need multisets and can just consider sets.

A basic net consists of

- a set of conditions $B$
- a set of events $E$
- a pre-condition map assigning a subset of conditions $\cdot e$ to any event $e$
- a post-condition map assigning a subset of conditions $e^\circ$ to any event $e$ such that $\cdot e \cup e^\circ \neq \emptyset$

The capacity of any condition is implicitly taken to be 1:

$$\forall b \in B : \text{Cap}(b) = 1$$

A marking $M$ is now a subset of conditions.

$$M \overset{e}{\rightarrow} M' \quad \text{iff} \quad \cdot q \subseteq M \quad \& \quad (M \setminus \cdot e) \cap e^\circ = \emptyset \quad \& \quad M' = (M \setminus \cdot e) \cup e^\circ$$
Concepts

Concurrency

Forwards conflict

Backwards conflict

Contact
Safe nets

- Contact occurs in marking $M$ if there exists an event $e$ such that
  \[ \bullet e \subseteq M \quad (M \setminus \bullet e) \cap e^\bullet \neq \emptyset \]

- A basic net is safe if there is no marking reachable from the initial marking in which contact occurs.
A safe Petri net semantics for CCS can be constructed by ‘surgery’ on the nets:

- Nil process
- Prefixing
- $p + q$
- $p \parallel q$