An important variant

A labelled transition system is a triple $(S, A, \rightarrow)$ where

- S is a set of states
- A is a set of labels (or actions, or operations, or events, ...)
- $\rightarrow \subseteq S \times A \times S$ $\quad(\rightarrow : S \to 2^{A \times S})$ transition relation

$$s \xrightarrow{a} s'$$

observable information about
what happens during the transition

particularly handy to model
communication / concurrency / distribution

Example An FSA $M = (Q, \Sigma, q_0, \delta, F)$ is an LTS:

$LTS_M = (S \cup \{\bullet\}, \Sigma \cup \{\checkmark\}, \rightarrow)$ where $s \rightarrow^{a} s' \iff a \in \Sigma \;\&\; s' \in \delta(s, a)$ or $a = \checkmark \;\&\; s' = \bullet \;\&\; s \in F$

$$\mathcal{L}_M = \{a_1 \ldots a_n \in \Sigma^* \mid \exists q_1, \ldots, q_n \mid q_0 \xrightarrow{a_1} \ldots q_{n-1} \xrightarrow{a_n} q_n \xrightarrow{\checkmark} \bullet\}$$

Communication-based concurrency

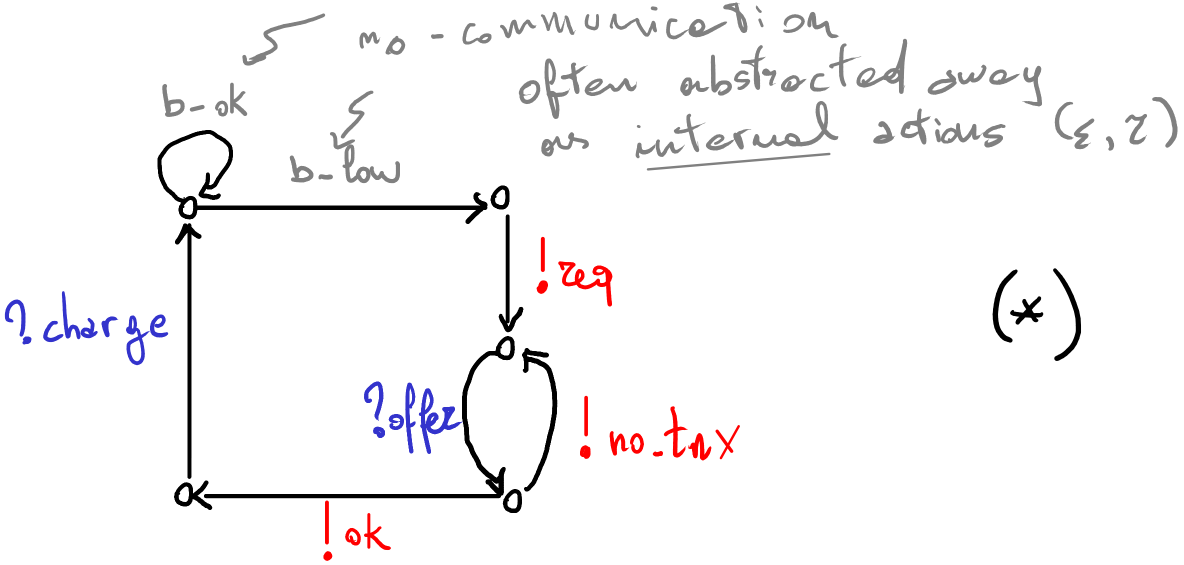
A robotic scenario:
Some mobile robots need to manage their energy in order to accoplish their task (e.g., patrolling some premises).

- When their batteries deplete, robots look for a recharge.
- Recharges are offered by recharge stations or other robots.

We can model this behaviour using an LTS capturing the observable features we are interested in: in this case communication
For instance, the behaviour of a robot seeking for a recharge is

The set of labels is the union of

- {b_low, b_ok} internal actions
- {?charge, ?offer} input actions
- {!req, !no_tnx} output actions



Exercise 4
Give an LTS modelling the behaviour of a robot offering a recharge.
Reflect about the "compatibility" between your solution and the LTS (*) above

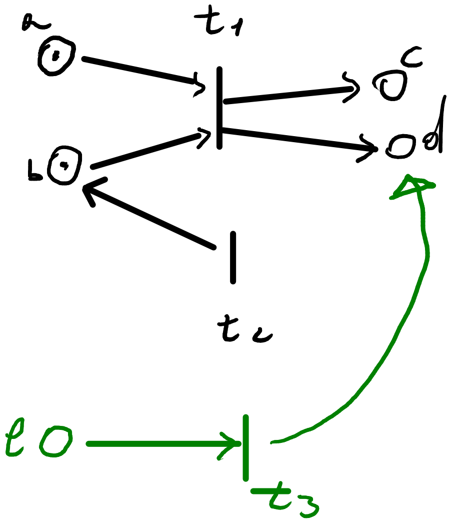
A more sophisticated example

Petri Nets

A Petri net (aka place-transition) net is a 4-tuple

$N = (P, T, F, m)$ where

- P is a finite set (of places)
- T is a finite set (of transitions)
- $F \subseteq (P \times T) \cup (T \times P)$ is a (flow) relation
- $m \subseteq P$ is the initial marking



with marking $m = \{a, b\}$

${}^{\bullet}t$ = {p in P | (p,t) in F}

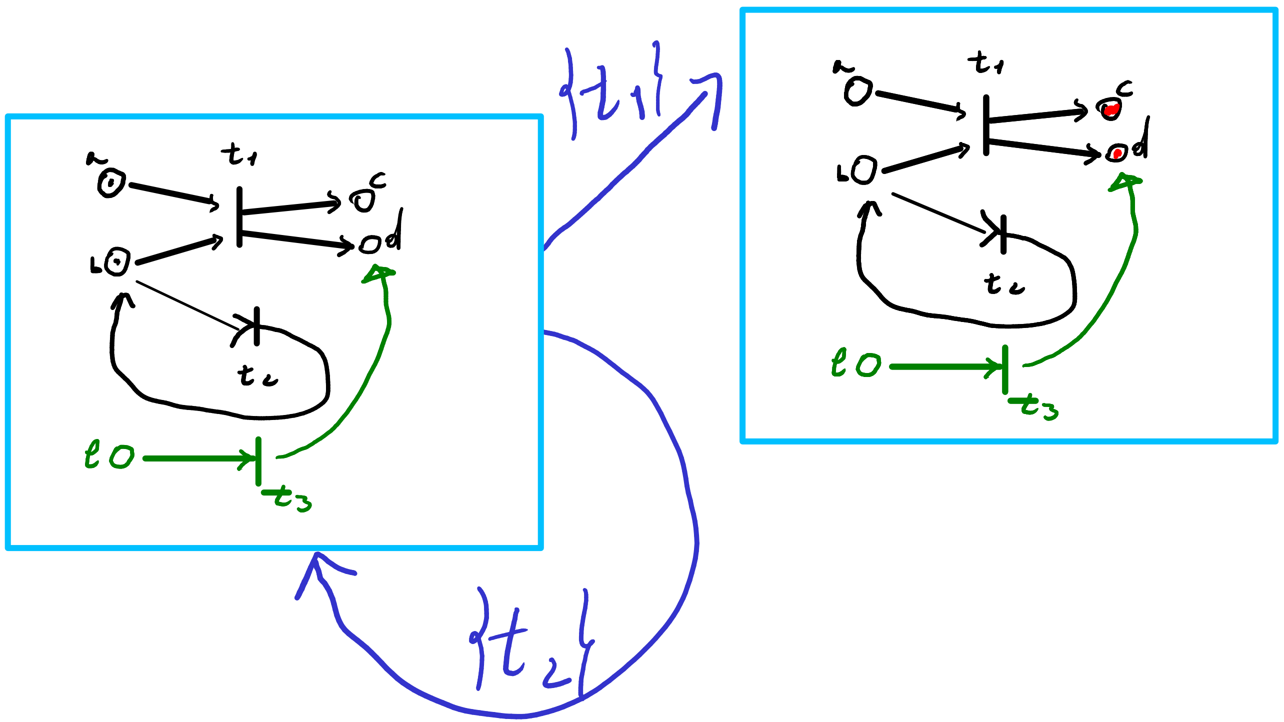
$t^{\bullet}$ = {p in P | (t,p) in F}

$t_1 \# t_2 \iff t_1 \neq t_2 \wedge {}^{\bullet}t_1 \cap {}^{\bullet}t_2 \neq \emptyset$

$$A = \{ X \subseteq T \mid \forall t, t' \in X : \text{not } t \# t' \}$$

$$m \xrightarrow{X} m' \quad \text{if} \quad {}^\bullet X \subseteq m \;\&\; m' = m \setminus {}^\bullet X \cup X^\bullet$$

$${}^\bullet X = \bigcup_{t \in X} {}^\bullet t \qquad X^\bullet = \bigcup_{t \in X} t^\bullet$$



Exercise 5
Give the transition system obtained from the initial marking {a,b,e}

Regular expressions

BNF-like syntax A, finite alphabet

$E ::= 0 \mid 1 \mid a \mid E+E \mid E\cdot E \mid E^*$, $a \in A$

end skip atomic instruct. if-then-else -;- iteration

Denotational semantics : $\mathcal{L} : E \longrightarrow 2^{A^*}$

Term-Algebra homomorphism

$\mathcal{L}(0) = \emptyset$ $\quad \mathcal{L}(1) = \{\varepsilon\}$ $\quad \mathcal{L}(a) = \{a\}$

$\mathcal{L}(E_1 + E_2) = \mathcal{L}(E_1) \cup \mathcal{L}(E_2)$

$\mathcal{L}(E_1 \cdot E_2) = \mathcal{L}(E_1) \cdot \mathcal{L}(E_2) \triangleq \{ vw \in A^* \mid v \in \mathcal{L}(E_1), w \in \mathcal{L}(E_2) \}$

$\mathcal{L}(E^*) = \mathcal{L}(E)^* = \bigcup_{n \geq 0} \mathcal{L}(E)^n$

Exercise 6
Prove or disprove that (a + b)* = (a* + b*)*

Term Algebras

assume $(\{f_1,\dots,f_n\}, ar)$ $\quad ar: \{f_1,\dots,f_n\} \to \omega$

$V \cap \{f_1,\dots,f_n\} = \emptyset$

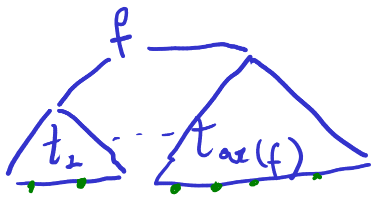
Term Algebra

The term algebra on a signature Σ and a countable set V of variables is the smallest set $Term_{\Sigma,V}$ s.t.

- $V \in T$
- $\forall f \in \Sigma,\ t_1,\dots,t_{ar(f)} \in Term_{\Sigma,V} : f(t_1,\dots,t_{ar(f)}) \in Term_{\Sigma,V}$

$T_\Sigma \subseteq Term_{\Sigma,V}$ is the set of closed terms

Exercise 7
Explain why in the above definition it is essential to require that $Term_{\Sigma,V}$ is the smallest set



- are either variables or "constants" (i.e $c \in \Sigma$ s.t $ar(c) = 0$)

Exercise 8
Give the term algebra for regular expressions

Transition System Specifications

"The first systematic study of TSSs may be found in [208], while the first study of TSSs with negative premises appeared in [57]." (Aceto et al.)

[208] R. d. Simone, Calculabilité et Expressivité dans l'Algèbre de Processus Parallèles Meije, thèse de 3 e cycle, Univ. Paris 7, 1984.
[57] B. Bloom, S. Istrail, and A. Meyer, Bisimulation can't be traced: preliminary report, in Conference Record 15th ACM Symposium on Principles of Programming Languages, San Diego, California, 1988, pp. 229–239. Preliminary version of Bisimulation can't be traced, J. Assoc. Comput. Mach., 42 (1995), pp. 232–268.

finite set of literals

A TSS with labels A is a set of (inference) rules $\dfrac{H}{\alpha}$

positive literals

LITERALS

positive

$\underbrace{t \xrightarrow{a} t' \quad \text{or} \quad t \in X}$

negative

$\underbrace{t \not\xrightarrow{a} \quad \text{or} \quad t \notin X}$

where $a \in A$, $t \in Term_{\Sigma,V}$, $X \subseteq Term_{\Sigma,V}$