

```

{*
*** Complete specification of the classical petri net method including execution semantics
*** (C) 2006-2014 by Manfred Jeusfeld (manfred.jeusfeld@acm.org); requires ConceptBase 7.1
*}
Place with
  attribute sendsToken: Transition;
  marks: Integer {* defines markings *}
end
Transition with attribute producesToken : Place
end
M in Function isA Integer with
  parameter p: Place
  constraint c1: $ (p marks this) $
end
Input in GenericQueryClass isA Place with
  parameter t: Transition
  constraint ci: $ (this sendsToken t) $
end
Enabled in QueryClass isA Transition with
  constraint c: $ forall p/Input[this] (M(p) > 0) $
end
ConnectedPlace in GenericQueryClass isA Place with
  parameter trans: Transition
  constraint c: $ (this sendsToken trans) or (trans producesToken this) $
end
inFlow in GenericQueryClass isA Transition!producesToken with
  parameter place: Place; trans: Transition
  constraint c1: $ From(this,trans) and To(this,place) $
end
outFlow in GenericQueryClass isA Place!sendsToken with
  parameter place: Place; trans: Transition
  constraint c1: $ From(this,place) and To(this,trans) $
end
{* Net effect of a transition t on a place p = incidence matrix.      *}
{* = number of links from t to p minus number of links from p to t  *}
IM in Function isA Integer with
  parameter p: Place; t: Transition
  constraint c1: $ (this = #inFlow[p,t] - #outFlow[p,t]) $
end
fireTransition in GenericQueryClass isA YesClass with
  parameter transition: Enabled
  constraint c1: $ (this=yes) $
end
{* Call fireTransition[t] for an enabled transition to execute the ECArule *}
ECArule UpdateConnectedPlaces with
  mode m: Deferred
  ecarule
    er : $ t/Transition p/Place m/Integer
      ON Ask fireTransition[t]
      IFNEW (t in Enabled) and (p in ConnectedPlace[t]) and
      (m = M(p)+IM(p,t))
      DO Retell (p marks m) $
end
{* that's all! *}

```