عنوان درس:

روش‌های رسمی در مهندسی نرم‌افزار

(Formal Methods in Software Engineering)

۲–۱– شبکه‌های پتری

دکتر محمّد عبداللّهی اَزگُمی

دانشیار گروه نرم‌افزار

دانشکده مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

azgomi@iust.ac.ir

فهرست مطالب

مقدمه

- تاکنون زبان‌های مدل‌سازی مختلفی معرفی شده‌اند. اما برای اهداف تحلیل خودکار تنها آنهایی مناسبند که دارای مبانی ریاضی یا صوری بوده و به اصطلاح صورت‌بندی (formalism) باشند.
- یک صورت‌بندی به یک زبان صوری و مبتنی بر نوعی از ریاضیات برای توصیف و بیان مدل‌ها گفته می‌شود.
- برخی از این صورت‌بندی‌ها، زبان‌های مدل‌سازی متنی (textual modeling language) هستند، مثل انواع جبرهای فرایندی (process algebras) نظیر:
  - فرآیندهای ارتباطی ترتیبی (CSP: Communicating Sequential Processes)
  - حساب سیستم‌های ارتباطی (CCS: Calculus of Communicating Systems)
- برخی دیگر، علاوه بر صوری بودن و داشتن روش بیان ریاضی، دارای قابلیت نمایش گرافیکی نیز هستند، نظیر شبکه‌های پتری (Petri nets).

شبکه‌های پتری

- مدل‌های ایجاد شده با صورت‌بندیها، برای تحلیل خودکار سیستم‌ها و به‌منظور ارزیابی جنبه‌های عملیاتی یا درستی‌یابی جنبه‌های کارکردی آنها مورد استفاده قرار می‌گیرند.
- موضوع بحث ما مدل‌های شبکه‌ای (net models) یا شبکه‌های پتری (Petri nets) است که در سال ۱۹۶۲ توسط کارل آدام پتری (Carl Adam Petri)، دانشمند آلمانی، برای مدل‌سازی سیستم‌های همروند (concurrent systems) معرفی شده است.

ویژگیهای شبکه‌های پتری

- **ویژگی اول: شبکه‌های پتری مبانی صوری (formal basis) دارند:**
  - **در حقیقت شبکه‌های پتری یک صورت‌بندی محسوب می‌شوند که صوری بودن یک نیازمندی کلیدی برای تحلیل خودکار مدل‌ها است.**
  - **شبکه‌های پتری یک رده از ماشین‌ها تحت عنوان اتوماتای شبکه پتری (Petri net automaton) را تعریف می‌کند.**
  - **تعریف صوری شبکه‌های پتری با استفاده از نظریه کیسه (bag theory) ارائه می‌شود.**
    - نظریه کیسه یا چندمجموعه (multiset) یک بسط نظریه مجموعه‌ها است که در آن هر کیسه، برخلاف مجموعه، می‌تواند اعضاء تکراری داشته باشد.

ویژگیهای شبکه‌های پتری

- **ویژگی دوم: شبکه‌های پتری نمادهای گرافیکی (graphical notations) دارند:**
  - **شبکه‌های پتری را می‌توان به‌صورت گرافیکی نمایش داد که این یک مزیت مهم برای فهم آسانتر مدل‌های ایجاد شده با شبکه‌های پتری است.**
- **این در حالی است که برخی از صورت‌بندی‌ها، نظیر جبرهای فرآیندی، گرافیکی نبوده و فقط یک زبان مدل‌سازی هستند.**
- **از طرف دیگر برخی زبان‌های مدل‌سازی گرافیکی، نظیر UML، اساساً صورت‌بندی محسوب نمی‌شوند.**

مفاهیم اولیه شبکه‌های پتری

- هر صورت‌بندی مدل‌سازی متشکل از حداقل دو مفهوم اولیه (primitive) است:
  - حالت (state)
  - کنش (action)
- در شبکه‌های پتری:
  - مفهوم اولیه مکان (place) برای توصیف حالت‌ها وجود دارد، و
  - مفهوم اولیه گذر (transition) برای مدل‌سازی کنش‌ها وجود دارد.

ساختار و رفتار شبکه‌های پتری

- مدل شبکه‌های پتری دارای یک ساختار ایستا (static structure) هستند که:
  - با استفاده از نظریه کیسه به‌طور صوری بیان می‌شود و
  - در عین حال قابلیت نمایش گرافیکی را با استفاده از گراف‌های شبکه‌های پتری (Petri net graphs) دارد.
- همچنین شبکه‌های پتری دارای یک رفتار پویا (dynamic behavior) نیز هستند، که:
  - نشانه‌گذاری شبکه‌های پتری (Petri net marking) و قواعد اجرای شبکه‌های پتری (Petri net execution rules) این رفتار را تعریف می‌کند.

تعریف صوری ساختار شبکه پتری

■ تعریف ۱: ساختار شبکه پتری (Petri net structure) یک پنج‌تایی $C = (P, T, I, O, H)$ است به نحوی که:

- □ P یک مجموعه متناهی از مکان‌ها است ($P = \{p_1, p_2, \ldots, p_n\}$).
- □ T یک مجموعه متناهی از گذرها است ($T = \{t_1, t_2, \ldots, t_m\}$)، به نحوی که P و T دو مجموعه مجزا هستند ($P \cap T = \varnothing$).
- □ $I: T \rightarrow Bag(P)$ یک تابع ورودی (input function) است که گذرها را به کیسه‌های مکان‌ها نگاشت می‌کند و در آن $Bag(P)$ مجموعه همه چندمجموعه‌های امکان‌پذیر P است.
- □ $O: T \rightarrow Bag(P)$ یک تابع خروجی (output function) است که گذرها را به کیسه‌های مکان‌ها نگاشت می‌کند.
- □ $H: T \rightarrow Bag(P)$ یک تابع بازدارنده (inhibition function) است که گذرها را به کیسه‌های مکان‌ها نگاشت می‌کند.

مثالی از یک مدل شبکه پتری

$C = (P, T, I, O, H)$

$P = \{p_1, p_2, p_3, p_4, p_5\}$

$T = \{t_1, t_2, t_3, t_4\}$

$I(t_1) = \{p_1\}$

$I(t_2) = \{p_2, p_3, p_5\}$

$I(t_3) = \{p_3\}$

$I(t_4) = \{p_4\}$

$O(t_1) = \{p_2, p_3, p_5\}$

$O(t_2) = \{p_5\}$

$O(t_3) = \{p_4\}$

$O(t_4) = \{p_2, p_3\}$

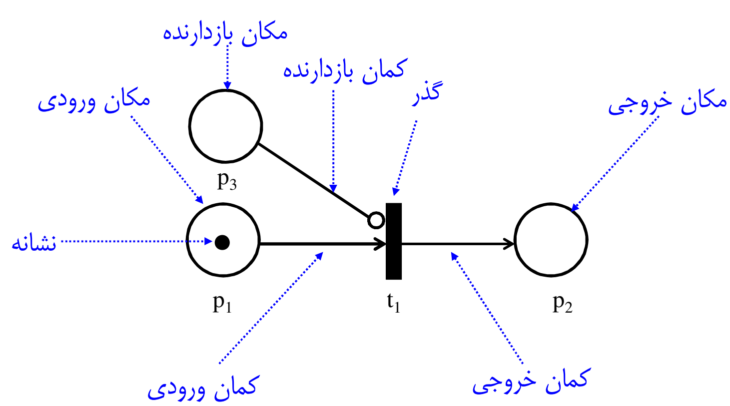
بازدارنده ندارد.

چند تعریف دیگر

- برای یک گذر $t \in T$، مجموعه مکان‌های ورودی را با $\bullet t$ (نقطه t)، مجموعه مکان‌های خروجی را با $t\bullet$ (t نقطه) و مجموعه مکان‌های بازدارنده را با ^{o}t (دایره t) نشان می‌دهیم که به‌صورت زیر تعریف می‌شوند:
  - $\bullet t = \{ p \in P : I(t, p) > 0 \}$
  - $t\bullet = \{ p \in P : O(t, p) > 0 \}$
  - $^{o}t = \{ p \in P : H(t, p) > 0 \}$
- با در نظر گرفتن Bag(P)، هر دو تعریف زیر درست خواهد بود:
  - I(t)، نشان‌دهنده چندمجموعه مکان‌های ورودی گذر t است.
  - I(t, p) نشان‌دهنده مضرب (multiplicity) (تعداد) عناصر p در چندمجموعه I(t) است.

نمایش گرافیکی شبکه‌های پتری

- اجزاء شبکه‌های پتری در نمایش گرافیکی در مثال ساده زیر نشان داده شده است:



نمایش گرافیکی شبکه‌های پتری

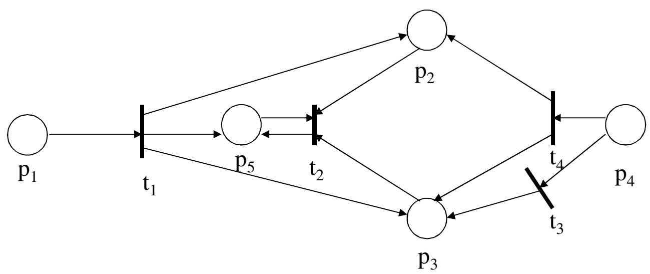
- **همانگونه که در شکل مشخص است، یک شبکه پتری متشکل از سه جزء اصلی است:**
  - **مکان‌ها (places) که با دایره نشان داده می‌شوند و حالت‌های امکان‌پذیر سیستم را مدل می‌کنند.**
  - **گذرها (transitions) که با مستطیل نشان داده می‌شوند و رخدادها یا کنش‌هایی هستند که باعث تغییر حالت‌ها می‌شوند.**
  - **کمان‌ها (arcs) که با پیکان نشان داده می‌شوند و مکان‌ها را به گذرها یا گذرها را به مکان‌ها متصل می‌کنند.**
- **در کنار سه جزء اصلی فوق، نشانه‌ها (token) هم وجود دارند که نشانه‌گذاری (marking) یا مقادیر قرار گرفته در مکان‌ها را مشخص می‌کنند.**

گراف‌های شبکه‌های پتری

- **نمایش گرافیکی شبکه‌های پتری برای تشریح مفاهیم نظریه شبکه پتری سودمند است.**
- **برای نمایش گرافیکی از گراف‌های شبکه‌های پتری استفاده می‌شود که یک گراف جهت‌دار دوسویه (bipartite directed multigraph) است. این گراف متشکل است از:**
  - **دو نوع گره:**
    - **مکان** که با دایره ( ) نشان داده می‌شود.
    - **گذر** که با مستطیل عمودی ( ) نشان داده می‌شود (البته در مراجع و ابزارهای مختلف نمادهای متفاوتی برای گذر استفاده می‌شود).
  - **کمان‌های جهت‌دار (directed arcs) که با پیکان نشان داده می‌شوند و مکان‌های ورودی را به گذرها یا گذرها را به مکان‌های خروجی متصل می‌کنند:**
    - کمان‌های ورودی و خروجی ممکن است که دارای یک برچسب عددی به عنوان **مضرب** (multiplicity) باشند: n
  - **کمان‌های بازدارنده (inhibitor arcs) که مکان‌های بازدارنده را به گذرها متصل می‌کنند:**
    - کمان‌های بازدارنده هم ممکن است که دارای مضرب باشند: m

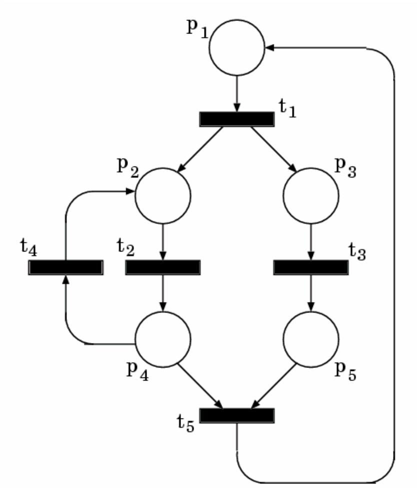
گراف‌های شبکه‌های پتری

- مثالی از گراف‌های شبکه‌های پتری در شکل زیر نشان داده شده است که معادل همان مثال ارائه شده با ساختار صوری است:



مثال دوم

نمایش گرافیکی | **ساختار صوری**



رفتار شبکه پتری

- همانگونه که گفته شد، شبکه‌های پتری علاوه بر ساختار صوری ایستا، دارای رفتار پویا هستند.
- از این نظر، شبکه‌های پتری مشابه برنامه‌ها هستند که دارای یک متن برنامه منبع (source code) بوده که به‌صورت ایستا است. اما یک برنامه پس از کامپایل شدن اجرا شده و رفتار پویایی را بروز می‌دهد.
- تعریف ساختار صوری یک شبکه پتری مثل یک متن برنامه منبع است. اما یک ابزار مدل‌سازی می‌تواند این تعریف ساختار را گرفته و شبکه پتری را اجرا کند.
- رفتار پویای شبکه‌های پتری با استفاده از سیستم شبکه پتری (Petri net system) تعریف می‌شود که مبتنی بر نشانه‌گذاری (marking) شبکه پتری و قواعد اجرای (execution rules) شبکه پتری است.

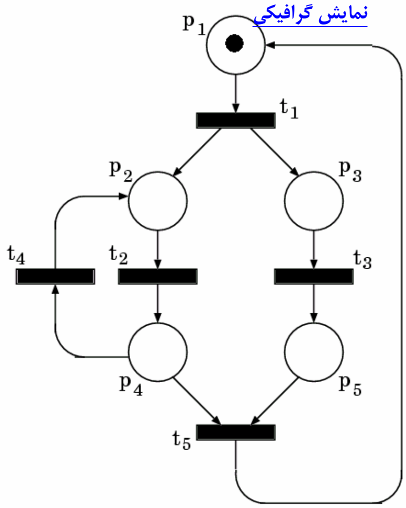
نشانه‌گذاری شبکه پتری

- نشانه‌گذاری (marking) همانند مکان و گذر یک مفهوم اولیه اصلی شبکه‌های پتری است.
- نشانه‌گذاری شبکه پتری انتساب نشانه‌ها (token) یا اعداد صحیح مثبت (N) به مکان‌های آن شبکه است.
- نشانه‌گذاری در تعریف ساختار صوری شبکه پتری با μ، M یا m نشان داده می‌شود.
- هر شبکه دارای یک نشانه‌گذاری اولیه (initial marking) است که با μ_0 ، M_0 یا m_0 نشان داده می‌شود.
- نشانه‌ها به مکان‌های شبکه پتری منتسب شده و می‌توانند آن باقی مانده یا در طی اجرای شبکه پتری تغییر کنند.
- در گراف شبکه پتری نشانه‌ها با دایره‌های کوچک توپر در داخل دایره مکان‌ها نشان داده می‌شوند: 
  - اگر تعداد نشانه‌ها در یک مکان زیاد باشد، عدد تعداد نشانه‌ها در داخل دایره مکان نوشته می‌شود: 12

مثال اول شبکه پتری نشانه‌گذاری شده

نمایش گرافیکی

ساختار صوری

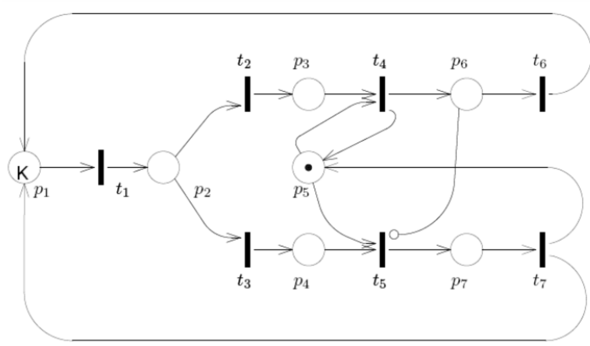


تعریف صوری سیستم شبکه پتری

- **تعریف ۲:** یک سیستم شبکه پتری یک شش‌تایی $C = (P, T, I, O, H, M_0)$ است به نحوی که:
  - که P، T، I، O و H مطابق تعریف (۱) هستند.
  - $M_0: P \rightarrow N$ تابع نشانه‌گذاری اولیه است که یک عدد صحیح مثبت را به هرکدام از مکان‌های شبکه پتری منتسب می‌کند.

مثال دوم شبکه پتری نشانه‌گذاری شده

- در این مثال t_5 دارای یک کمان بازدارنده بوده و مکان p_1دارای k نشانه است.



$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$
$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$

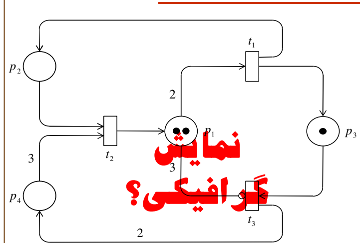
$I(t_1) = \{p_1\}$
$I(t_2) = \{p_2\}$
$I(t_3) = \{p_2\}$
$I(t_4) = \{p_3, p_5\}$
$I(t_5) = \{p_4, p_5\}$
$I(t_6) = \{p_6\}$
$I(t_7) = \{p_7\}$

$O(t_1) = \{p_2\}$
$O(t_2) = \{p_3\}$
$O(t_3) = \{p_4\}$
$O(t_4) = \{p_5, p_6\}$
$O(t_5) = \{p_7\}$
$O(t_6) = \{p1\}$
$O(t_7) = \{p1, p5\}$

$H(t_5) = \{p_6\}$

$M_0 = (k, 0, 0, 0, 1, 0, 0)$

مثال سوم شبکه پتری نشانه‌گذاری شده



- توصیف صوری شبکه پتری:

$N = (P, T, I, O, H)$

$P = \{p_1, p_2, p_3, p_4\}$

$T = \{t_1, t_2, t_3\}$

- تابع ورودی: $I(t_2) = \{p_2\}, I(t_2) = \{p'_4 3\}, I(t_1) = \{p'_1 2\}, I(t_3) = \{p_3\}$

برای سایر موارد $I(t_j) = \emptyset$

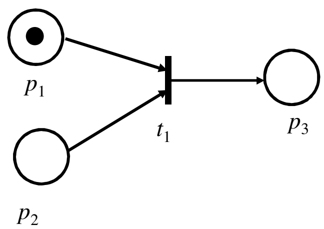
- تابع خروجی: $O(t_2) = \{p_1\}, O(t_1) = \{p_2\}, O(t_1) = \{p_3\}, O(t_3) = \{p'_4 2\}$

برای سایر موارد $O(p_i) = \emptyset$

- تابع بازدارندگی: $H(t_3) = \{p'_1 3\}$ ..... برای سایر موارد $H(t_j) = \emptyset$.
- نشانه‌گذاری اولیه: $M_0=(2, 0, 1, 0)$

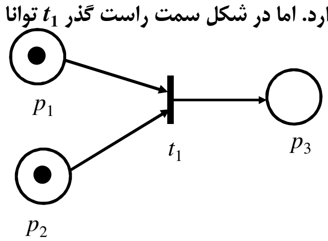
توانا بودن گذر

- به‌صور غیرصوری، یک گذر توانا (enabled) یا قابل شلیک (firable) است اگر:
  1. در هر کدام از مکان‌های ورودی حداقل به تعداد یا مضرب کمان‌های ورودی، نشانه وجود داشته باشد.
  2. در مکان‌های متصل با کمان‌های بازدارنده کمتر از تعداد یا مضرب کمان‌های بازدارنده نشانه وجود داشته باشد.
  3. گذر فاقد کمان‌های ورودی و بازدارنده: همیشه توانا است.
- برای مثال در دو شکل زیر، در شکل سمت چپ گذر t_1 توانا نیست چون در مکان p_2 هیچ نشانه‌ای وجود ندارد. اما در شکل سمت راست گذر t_1 توانا است.



t_1 توانا نیست:

$M(p_1)\geq 1$ & $M(p_2)\geq 1$ = false



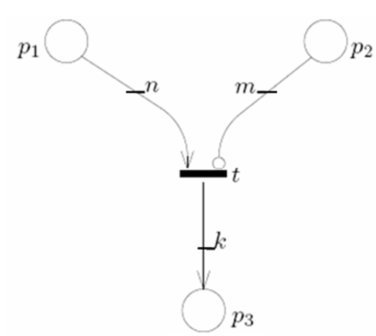
t_1 توانا است:

$M(p_1)\geq 1$ & $M(p_2)\geq 1$ = true

توانا بودن گذر

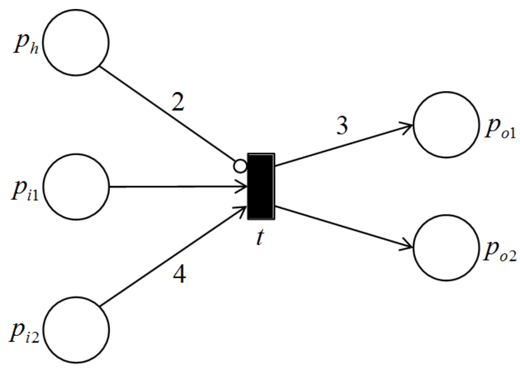
- در مثال زیر گذر t در صورتی توانا خواهد بود که در مکان p_1 به تعداد n یا بیشتر از آن نشانه وجود داشته باشد و در مکان p_2 کمتر از m نشانه وجود داشته باشد.
- به‌طور صوری توانا بودن t به‌صورت زیر بیان می‌شود:

$$M(p_1) \geq n \ \& \ M(p_2) < m$$



توانا بودن گذر

- مثال دیگر:



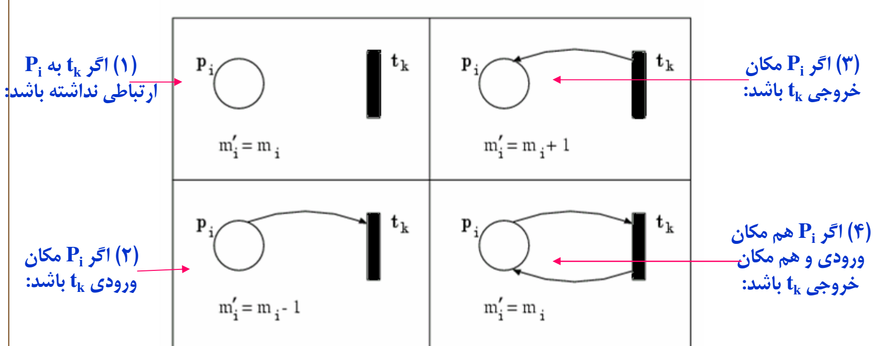
- شرط توانا شدن؟

$$m(p_{i1}) \geq 1$$
$$m(p_{i2}) \geq 4$$
$$m(p_h) < 2$$

قواعد اجرای شبکه پتری

- اگر یک گذر در نشانه‌گذاری **m** توانا باشد، اجرا می‌شود که اصطلاحاً گفته می‌شود که شلیک می‌کند (**fire**) و شبکه را به یک نشانه‌گذاری جدید **m'** می‌برد. برای این منظور چهار قاعده وجود دارد که در شکل زیر برای گذر t_k آمده است:

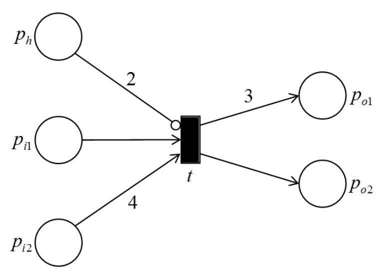
m → t_k → m'



قواعد اجرای شبکه پتری

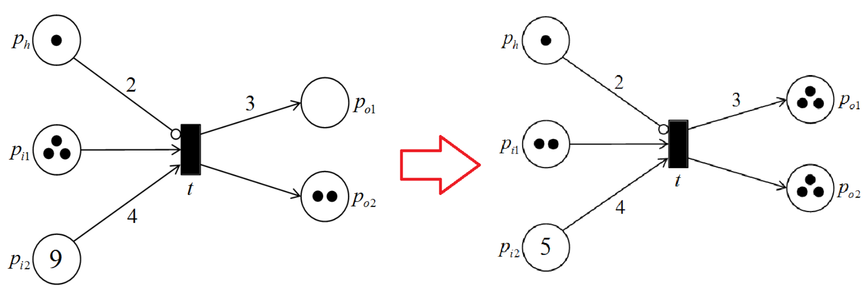
به بیان دیگر، وقتی یک گذر اجرا شود:

- از هر مکان ورودی مثل p_i به تعداد برچسب کمان مربوطه، $I(p_i, t)$ نشانه برداشته می‌شود.
- به هر مکان خروجی مثل p_o به تعداد برچسب کمان مربوطه، $O(t, p_o)$ نشانه اضافه می‌شود.



$$m(p_{i1}) \ -= 1$$
$$m(p_{i2}) \ -= 4$$
$$m(p_{o1}) \ += 3$$
$$m(p_{o2}) \ += 1$$

قواعد اجرای شبکه پتری



$$m\ [t\rangle\, m'$$

تعریف صوری توانا بودن و شلیک کردن گذر

- **تعریف ۳:** گذر t در یک نشانه‌گذاری M توانا است اگر و فقط اگر داشته باشیم:

$$\bullet\ \forall p \in {}^{\bullet}t,\ M(p) \ge I(t,p) \qquad and$$

$$\bullet\ \forall p \in {}^{\circ}t,\ M(p) < H(t,p)$$

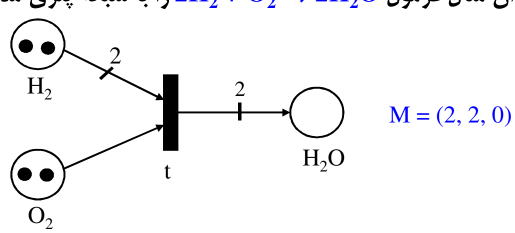
- وقتی t توانا بوده و شلیک کند:
  - از مجموعه مکان‌های ورودی ($\bullet t$) به تعداد مضرب کمان متصل‌کننده آن مکان به t نشانه حذف می‌کند.
  - به مجموعه مکان‌های خروجی ($t\bullet$) به تعداد مضرب کمان متصل‌کننده t به آن مکان نشانه اضافه می‌کند.
- **تعریف ۴:** شلیک کردن گذر t در نشانه‌گذاری M که در آن توانا است باعث ایجاد نشانه‌گذاری M' می‌شود، به نحوی که:

$$M' = M + O(t) - I(t)$$

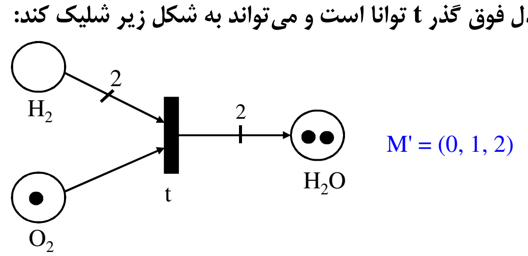
  - به اختصار شلیک کردن گذر t را با $M\,[t\rangle\,M'$ نشان می‌دهیم.

مثالی از شلیک کردن یک گذر

- به عنوان مثال فرمول $2H_2 + O_2 \rightarrow 2H_2O$ را با شبکه پتری مدل می‌کنیم:



- در مدل فوق گذر t توانا است و می‌تواند به شکل زیر شلیک کند:



مثالی از شلیک کردن یک گذر

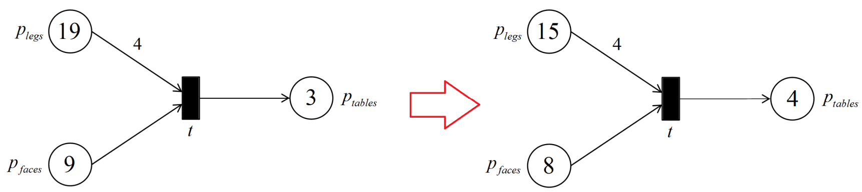
نشانه‌گذاری اولیه $m_0 = (19, 9, 3)$

$m_1 = (15, 8, 4)$

$m_2 = (11, 7, 5)$

$m_3 = (7, 6, 6)$

$m_4 = (3, 5, 7)$



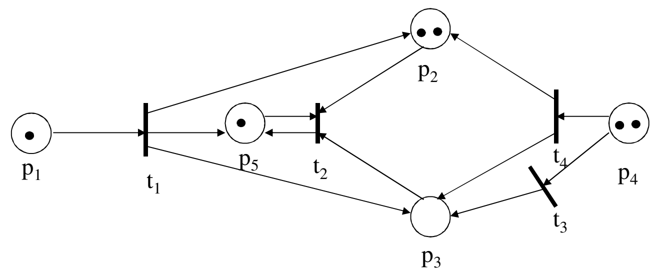
$m_0\ [t\rangle\ m_1\ [t\rangle\ m_2\ [t\rangle\ m_3\ [t\rangle\ m_4$

اجرای شبکه پتری

- اجرای شبکه پتری بوسیله تعداد نشانه‌ها در مکان‌های شبکه پتری کنترل می‌شود.
- شلیک کردن یا کامل شدن (completion) گذرها، نشانه‌های موجود در مکان‌ها را کم و زیاد می‌کند.
- اجرای شبکه پتری به‌صورت غیرقطعی (non-deterministic) است. این بدان معنی است که:
  - چندین گذر ممکن است که در یک زمان (یا در یک نشانه‌گذاری) توانا باشند که یکی از آنها می‌تواند شلیک می‌کند.
  - هر کدام از گذرها ممکن است در یک زمانی بین صفر تا بی‌نهایت شلیک کنند.
  - انتخاب هرکدام از گذرها برای شلیک کردن به‌طور غیرقطعی انجام می‌شود.
- چون شلیک کردن گذرها به‌طور غیرقطعی است، شبکه‌های پتری برای مدل‌سازی رفتار همروند (concurrent) در سیستم‌های توزیع‌شده (distributed) مناسب هستند.

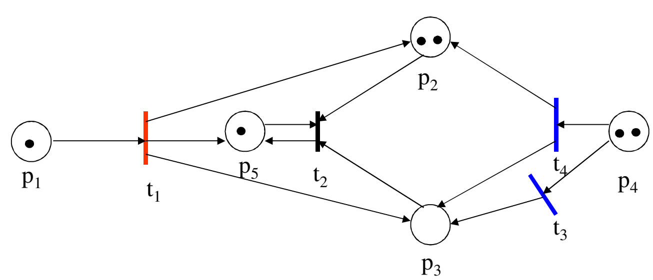
مثالی از اجرای یک شبکه پتری

- حالا همان مثال اولیه را در نظر می‌گیریم و با نشانه‌گذاری شبکه را در چند مرحله اجرا می‌کنیم:
  - نشانه گذاری اولیه: $M_0 = (1, 2, 0, 2, 1)$



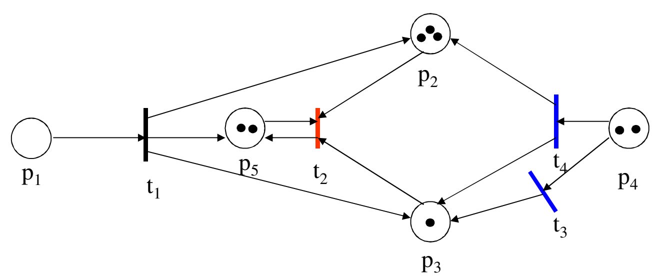
مثالی از اجرای یک شبکه پتری

- در نشانه‌گذاری اولیه گذرهای t_1، t_3 و t_4 توانا هستند، که یکی از آنها به‌طور غیرقطعی شلیک می‌کند.
- فرض می‌کنیم که t_1 شلیک کند:



مثالی از اجرای یک شبکه پتری

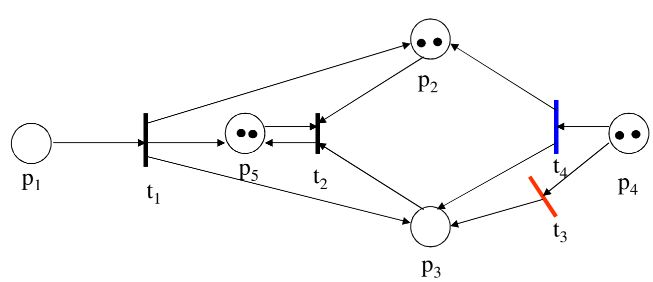
- پس از شلیک کردن گذر t_1 نشانه گذاری جدید $M_1 = (0, 3, 1, 2, 2)$ ظاهر می‌شود:



- در نشانه‌گذاری M_1 گذرهای t_2، t_3 و t_4 توانا هستند، که فرض می‌کنیم که t_2 شلیک می‌کند...

مثالی از اجرای یک شبکه پتری

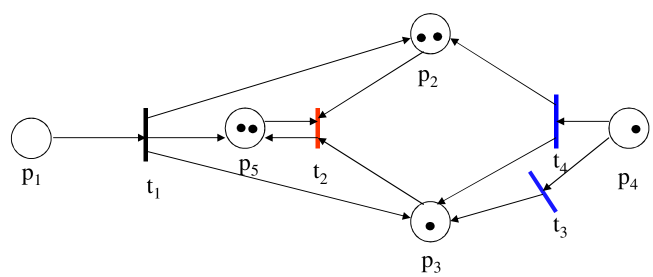
- پس از شلیک کردن گذر t_2 نشانه گذاری جدید $M_2 = (0, 2, 0, 2, 2)$ ظاهر می‌شود:



- در نشانه‌گذاری M_2 گذرهای t_3 و t_4 توانا هستند، که فرض می‌کنیم که t_3 شلیک می‌کند...

مثالی از اجرای یک شبکه پتری

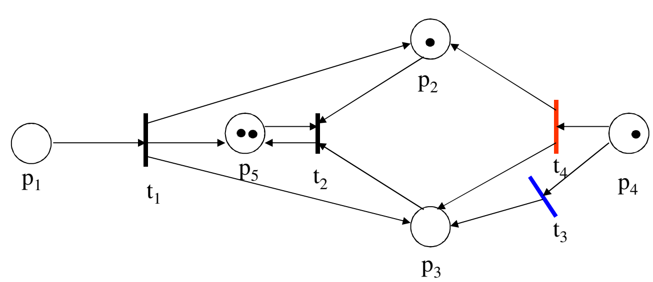
- پس از شلیک کردن گذر t_2 نشانه گذاری جدید $M_3 = (0, 2, 1, 1, 2)$ ظاهر می‌شود:



- در نشانه‌گذاری M_3 گذرهای t_2، t_3 و t_4 توانا هستند، که فرض می‌کنیم که t_2 شلیک می‌کند...

مثالی از اجرای یک شبکه پتری

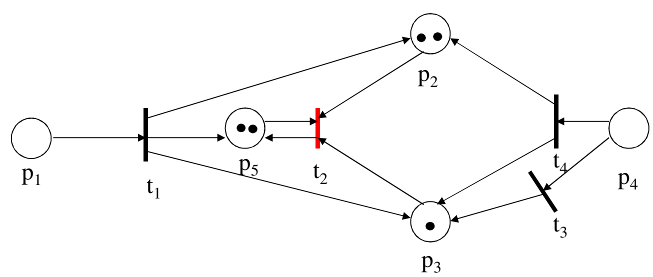
- پس از شلیک کردن گذر t_2 نشانه گذاری جدید $M_4 = (0, 1, 0, 1, 2)$ ظاهر می‌شود:



- در نشانه‌گذاری M_4 تنها گذر t_3 و t_4 توانا است، که فرض می‌کنیم t_4 شلیک می‌کند...

مثالی از اجرای یک شبکه پتری

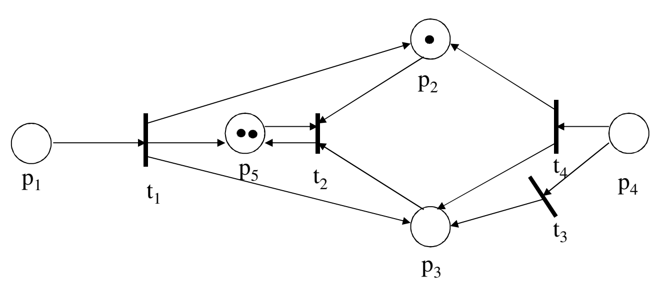
- پس از شلیک کردن گذر t_4 نشانه گذاری جدید $M_5 = (0, 2, 1, 0, 2)$ ظاهر می‌شود:



- در نشانه‌گذاری M_5 تنها گذر t_2 توانا است که شلیک می کند...

مثالی از اجرای یک شبکه پتری

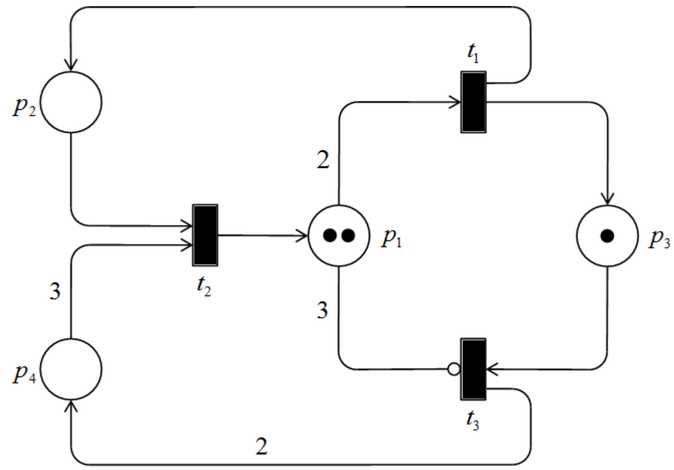
- پس از شلیک کردن گذر t_2 نشانه گذاری جدید $M_6 = (0, 1, 0, 0, 2)$ ظاهر می‌شود:



- در نشانه‌گذاری M_6 هیچ گذری توانا نیست و اجرای شبکه در اینجا متوقف می‌شود.
- البته در حالت کلی مدل‌های سیستم‌های همروند به این صورت پس از چند مرحله اجرا خاتمه پیدا نمی‌کنند، بلکه به‌صورت خاتمه‌ناپذیر (non-terminating) هستند.

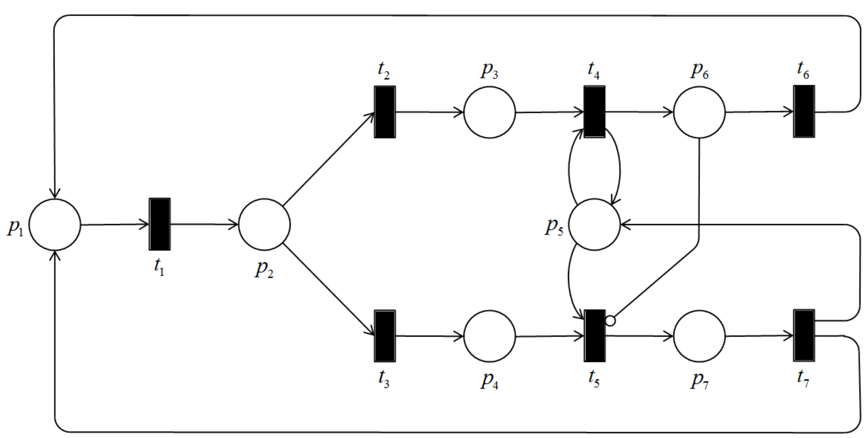
تمرین ۱: توصیف صوری مدل

- توصیف صوری مدل شبکه پتری زیر را بنویسید:



تمرین ۲: توصیف صوری مدل

- توصیف صوری مدل شبکه پتری زیر را بنویسید:



تمرین ۳: مدل شبکه پتری ساعت

- یک مدل شبکه پتری برای مدل‌سازی ساعت ارائه کنید، طوری که مکانسیم ساعت را برای ۲۴ ساعت شبانه‌روز نشان دهد:
  - ۶۰ ثانیه یک دقیقه
  - ۶۰ دقیقه یک ساعت
  - ۲۴ ساعت یک شبانه روز
- محدودیت این مدل شبکه پتری در زمینه مدل‌سازی زمان واقعی چیست؟

تمرین ۴: کار با ابزار شبکه پتری

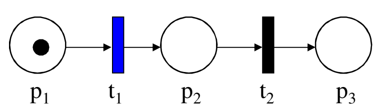
- هر سه مدل تمرین‌های قبل را با یک ابزار شبکه پتری ساخته و اجرا (قابلیت انیمیشن یا بازی نشانه‌ها) کنید. مدل ساخته شده را تحویل دهید.

مدل‌سازی سیستم‌های همروند با شبکه‌های پتری

- همانگونه که گفته شد شبکه‌های پتری برای مدل‌سازی سیستم‌های همروند مناسب هستند. در حقیقت شبکه‌های پتری همانند جبرهای فرآیندی و برخی روشهای صوری دیگر، جزء مدل‌های همروندی (concurrency models) محسوب می‌شوند.
- در این بخش نحوه مدل‌سازی مفاهیم مهم سیستم‌های همروند با شبکه‌های پتری را ارائه می‌کنیم. مفاهیمی که به آنها پرداخته می‌شود عبارتند از:
  - کنش‌های ترتیبی (sequential actions)،
  - همگام‌سازی کنش‌ها (synchronization)،
  - ادغام کنش‌ها (merging)،
  - همروندی کنش‌ها (concurrency)،
  - تعارض کنش‌ها (conflict)، و
  - منابع محدود (limited resources).
- با ارائه زیرمدل‌هایی برای موارد فوق، بلوک‌های پیش‌ساخته‌ای (building blocks) فراهم می‌شود که با استفاده از آنها می‌توان اغلب سیستم‌های همروند را با شبکه‌های پتری مدل‌سازی نمود.

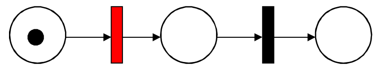
مدل‌سازی کنش‌های ترتیبی

- در یک سیستم همروند ممکن است که دو کنش a_1 و a_2 وجود داشته باشند که a_1 پیش‌نیاز a_2 باشد. یعنی این دو کنش باید به‌صورت ترتیبی اجرا شوند، نخست a_1 و سپس a_2. (مدل‌سازی روابط علت و معلولی)
- برای مدل‌سازی دو کنش فوق می‌توانیم دو گذر t_1 و t_2 را به‌ترتیب متناظر با a_1 و a_2 در نظر بگیریم و مدل زیر را ایجاد کنیم که متشکل از این دو گذر و سه مکان p_1، p_2 و p_3 است:

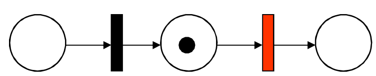


- در این مدل مکان p_1 نشانه‌گذاری شده است و لذا گذر t_1 توانا است. اما مکان p_2 خالی بوده و t_2 توانا نیست. بنا بر این اگر گذر t_1 شلیک کند یک نشانه در p_2 گذاشته شده و t_2 توانا می‌شود. بنا بر این اجرای t_2 پس از t_1 تضمین شده است...

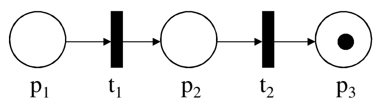
مدل‌سازی کنش‌های ترتیبی



مدل‌سازی کنش‌های ترتیبی

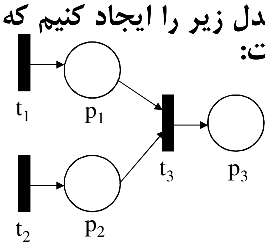


مدل‌سازی کنش‌های ترتیبی



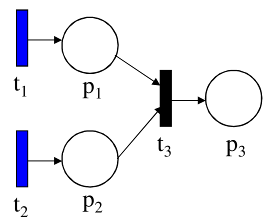
مدل‌سازی همگام‌سازی کنش‌ها

- در خیلی از مواقع در یک سیستم همروند اجرای دو کنش a_1 و a_2 پیشنیاز اجرای کنش دیگری مثل a_3 است. یعنی تا a_1 و a_2 کامل نشوند، a_3 نمی‌تواند شروع شود. به‌عبارت دیگر a_3 باید خود را با اجرای دو کنش دیگر **همگام‌سازی** (synchronize) نماید.
- برای مدل‌سازی همگام‌سازی a_3 با a_1 و a_2 می‌توانیم به ترتیب سه گذر t_3، t_1 و t_2 را در نظر بگیریم و مدل زیر را ایجاد کنیم که متشکل از این سه گذر و سه مکان p_1، p_2 و p_3 است:

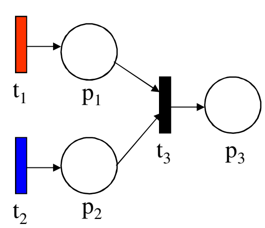


- در این مدل تنها پس از کامل شدن هر دو گذر t_1 و t_2 گذر t_3 توانا شده و می‌تواند شلیک کند.

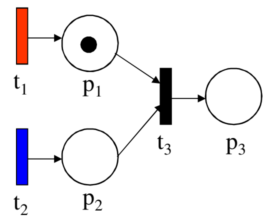
مدل‌سازی همگام‌سازی کنش‌ها



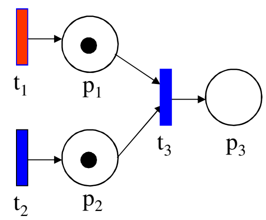
مدل‌سازی همگام‌سازی کنش‌ها



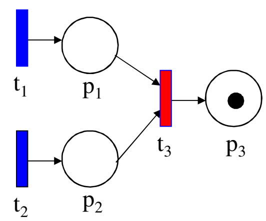
مدل‌سازی همگام‌سازی کنش‌ها



مدل‌سازی همگام‌سازی کنش‌ها

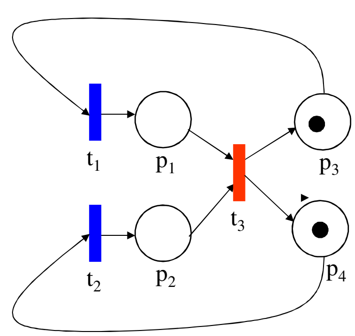


مدل‌سازی همگام‌سازی کنش‌ها



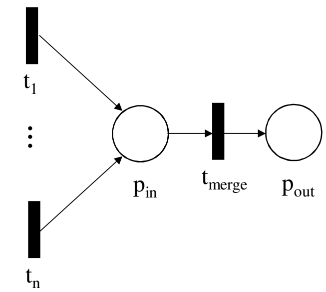
مدل‌سازی همگام‌سازی کنش‌ها

- در صورتی که بخواهیم اجرای t_1 و t_2 یک بار به ازای هر اجرای t_3 انجام شود می‌توانیم مدل زیر را درست کنیم، که در آن در نشانه‌گذاری اولیه t_1 و t_2 توانا هستند و پس از اجرای آنها t_3 توانا می‌شود:



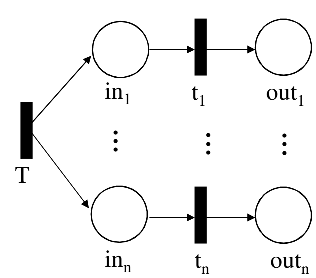
مدل‌سازی ادغام کنش‌ها

- ممکن است که نتایج اجرای چند کنش با هم ادغام شوند تا از سرویسهای یک کنش دیگر استفاده کنند. در این صورت می‌توانیم مدلی مثل شکل زیر را ایجاد کنیم:

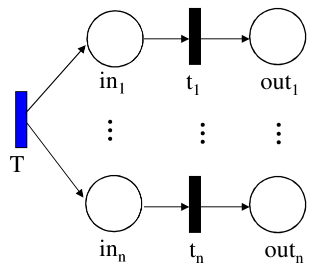


مدل‌سازی کنش‌های همروند

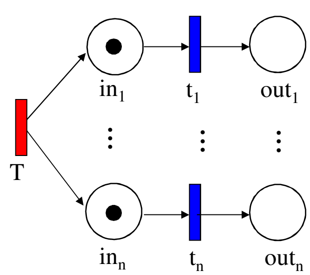
- ماهیتاً در سیستم‌های همروند کنش‌هایی که باید به‌طور همزمان اجرا شوند داریم. یعنی پس از کامل شدن یک کنش که منجر به برآورده شدن یک شرط می‌شود، چند کنش بتوانند به‌طور همزمان اجرا شوند.
- برای این منظور می‌توانیم مدلی مثل شکل زیر را ایجاد کنیم که در آن با کامل شدن گذر T، گذرهای t_1 الی t_n توانا شده و می‌توانند به‌طور همزمان اجرا شوند...



مدل‌سازی کنش‌های همروند

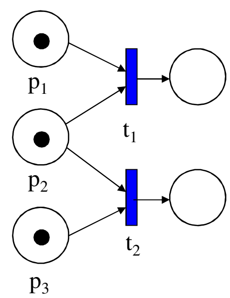


مدل‌سازی کنش‌های همروند

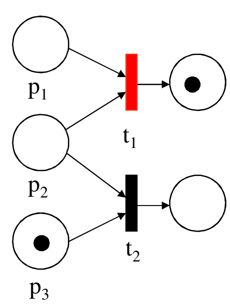


مدل‌سازی تعارض کنش‌ها

- تعارض یا تضاد (conflict) هم ممکن است که ماهیتاً در سیستم‌های همروند وجود داشته باشد و لزوماً اتفاق بدی محسوب نمی‌شود. یعنی دو یا چند کنش داشته باشیم که اجرای یکی منجر به ناتوان شدن کنش‌های دیگر شود.
- برای مثال در مدل زیر t1 و t2 را متناظر با دو کنش دارای تعارض در نظر گرفته‌ایم. مکان p2 متناظر با یک منبع مشترک است. در صورتی که این منبع توسط کنش a1 استفاده شود، کنش a2 دیگر نمی‌تواند کامل شود. در این مدل با کامل شدن گذر t1، نشانه از مکان p2 برداشته شده و باعث می‌شود که گذر t2 ناتوان شود...



مدل‌سازی تعارض کنش‌ها

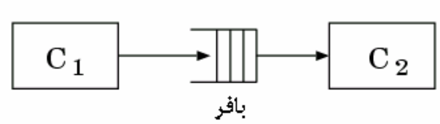


مدل‌سازی تعارض کنش‌ها

- اما گاهی اوقات تعارض تبعاتی نظیر بن‌بست (deadlock) را در پی دارد که به این دلیل ممکن است که اتفاق ناخوشایندی باشد و لذا باید تشخیص داده شده و برطرف شود:
  - برای برطرف کردن تعارض ممکن است که کنش‌ها را به نحوی مناسب پس و پیش کنیم.
  - یا آنکه با یک روش احتمالی عمل کنیم و در صورت بروز تعارض یکی از دو کنش به‌طور احتمالی کامل شوند.

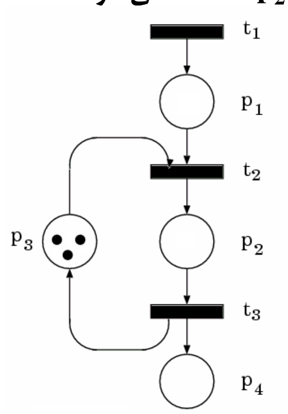
مدل‌سازی منابع محدود

- یکی دیگر از مواردی که در مدل‌سازی سیستم‌های همروند لازم است، منابع محدود هستند.
- برای مثال در شکل زیر فرآیند C_1 خروجی‌هایی را تولید نموده و در بافر دارای ظرفیت سه قرار می‌دهد که توسط فرآیند C_2 مصرف می‌شود.



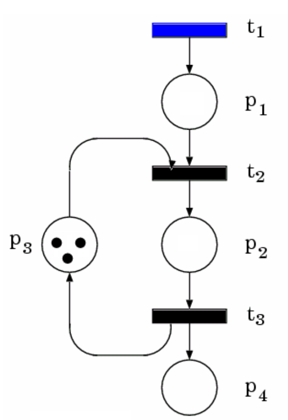
مدل‌سازی منابع محدود

- برای مدل‌سازی سیستم فوق می‌توانیم مدل زیر را درست کنیم. در این مدل p_2 متناظر با بافر در نظر گرفته شده است. همچنین از مکان p_3 برای کنترل ظرفیت یا تعداد نشانه‌های p_2 استفاده می‌شود.



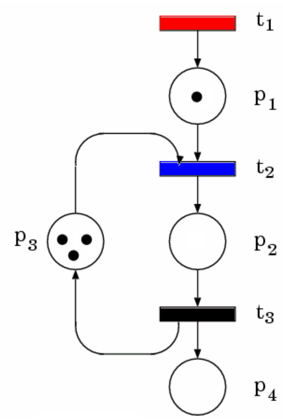
مدل‌سازی منابع محدود

- در این مدل t_1 توانا است و می‌تواند شلیک کند...



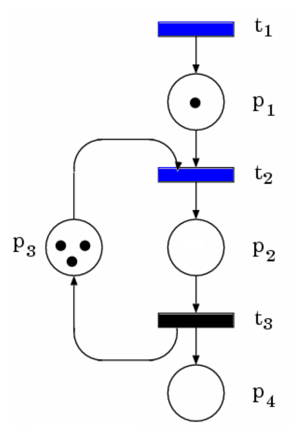
مدل‌سازی منابع محدود

- پس از شلیک کردن t_1 گذر t_2 توانا می‌شود...



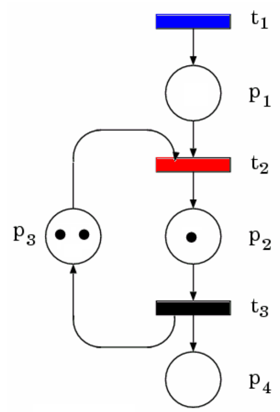
مدل‌سازی منابع محدود

- در عین حال t_1 هم هنوز توانا است. ممکن است هر کدام از دو گذر t_1 یا t_2 شلیک کنند. اما فرض می‌کنیم که t_2 شلیک کند...



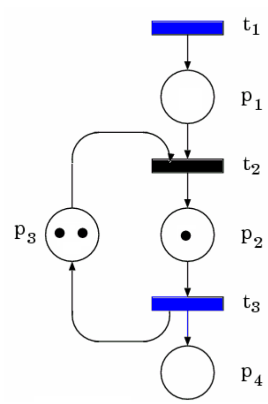
مدل‌سازی منابع محدود

- با شلیک کردن t_2 یک نشانه از هر کدام از دو مکان p_1 و p_3 برداشته می‌شود...



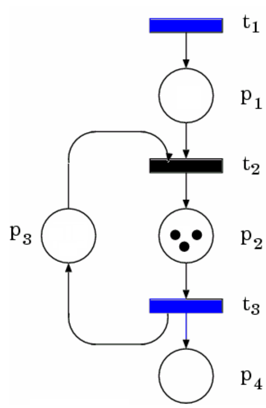
مدل‌سازی منابع محدود

- ممکن است که شلیک کردن‌های متناوب t_1 و t_2 چند بار دیگر و قبل از شلیک کردن t_3 ادامه یابد. اما شلیک کردن t_2 نمی‌تواند بیش از سه بار، قبل از شلیک کردن t_3 ادامه یابد...



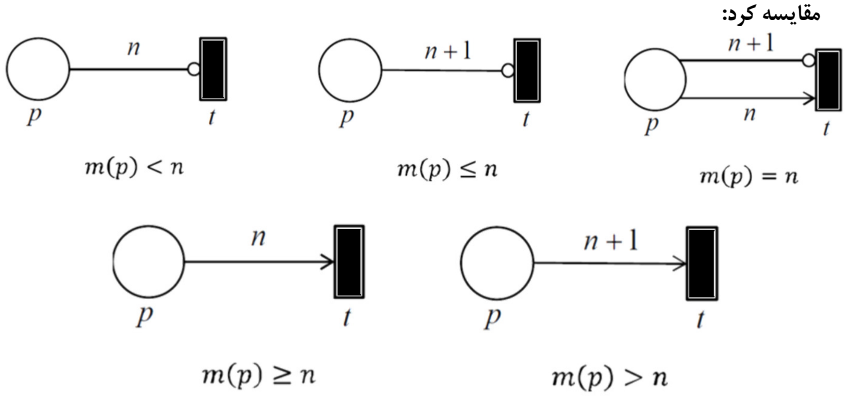
مدل‌سازی منابع محدود

- زیرا مکان p_3 خالی شده و نمی‌تواند بیش از اجرا شود. در این وضعیت سه نشانه در مکان p_2 قرار گرفته و بدین ترتیب ظرفیت مکان p_3 کنترل می‌شود.



مدل‌سازی شرط‌های منطقی

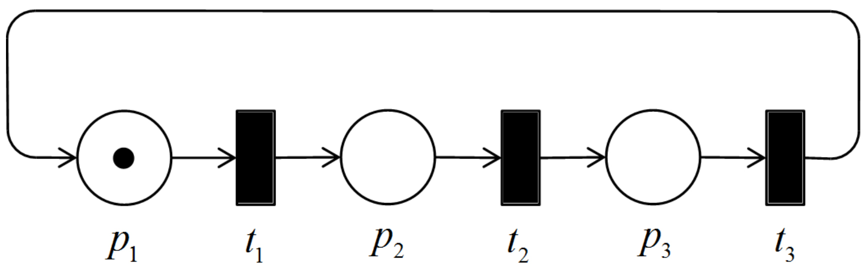
- می‌توان نشانه‌گذاری یک مکان را با عملگرهای مقایسه‌ای $<$، $\leq$، $=$، $\geq$ و $>$ را با یک عدد مشخص مقایسه کرد:



- مقایسه نشانه‌گذاری یک مکان با یک عدد بدون اینکه نشانه‌گذاری یک مکان تغییر کند
  - می‌توان با استفاده از کمان‌های خروجی نشانه‌های برداشته شده را دوباره به مکان p برگرداند.

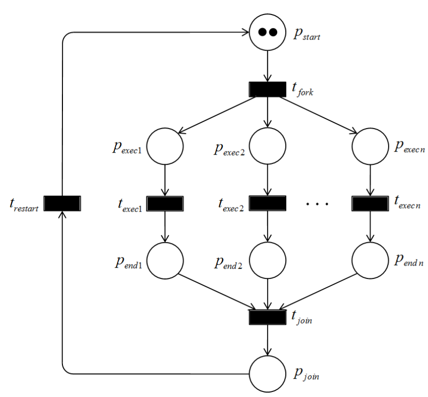
مدل‌سازی ساختارهای تکرار

- فرآیندی که سه فعالیت
  - تولید یک فایل داده (t_1)
  - فشرده‌سازی آن (t_2)
  - ارسال آن (t_3)
- را مرتباً تکرار می‌کند:



مدل‌سازی ساختارهای انشعاب و ادغام

- رفتارهای انشعاب و ادغام در سیستم‌های توزیع شده
  - گذر t_{fork}: انشعاب
  - گذر t_{join}: همگام‌سازی مربوط به پایان انشعاب
- وقتی گذر های مصرف شده در آن توسط یکبار شود لزوماً به این معنی نیست که نشانه‌ی‌اجرا می t_{join} اجرای گذر t_{fork} تولید شده‌اند

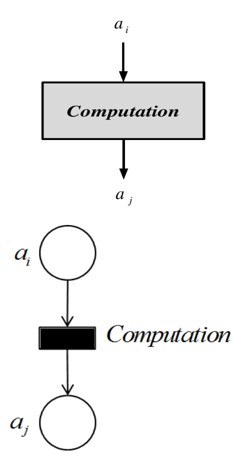


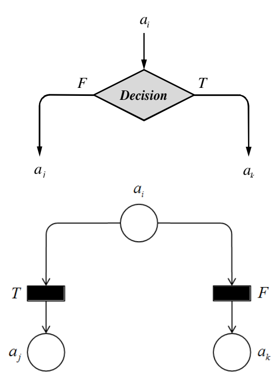
مدل‌سازی روندنما (فلوچارت)

- دو جنبه از فرآیند
  - محاسبات
  - جریان کنترلی
    - ترتیب انجام محاسبات
- توصیف ساختار جریان کنترلی یک برنامه
- روندنما (فلوچارت) نه محاسباتی که قرار است انجام شود، بلکه ساختار برنامه و ترتیب انجام محاسبات را توصیف می‌کند.

مدل‌سازی روندنما (فلوچارت)

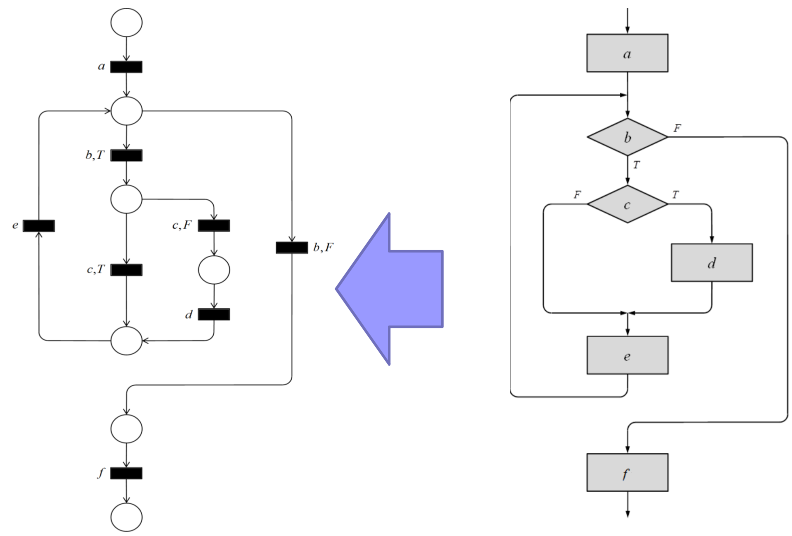
- برای این منظور، در شبکه پتری مربوطه هیچ اطلاعاتی راجع به متغیرهای برنامه نگهداشته نمی‌شود.
- از بین گذرهای توانای موجود کدام اجرا می‌شوند؟
  - کاملاً غیرقطعی





نگاشت الگوهای موجود در روندنما به شبکه‌های پتری

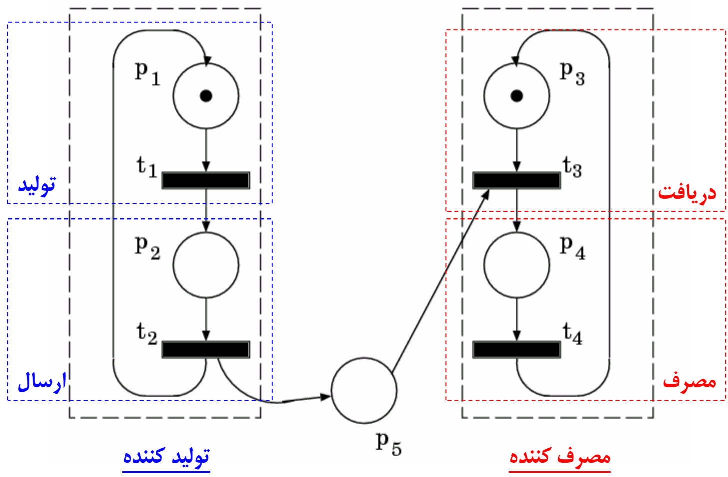
مدل‌سازی روندنما (فلوچارت)



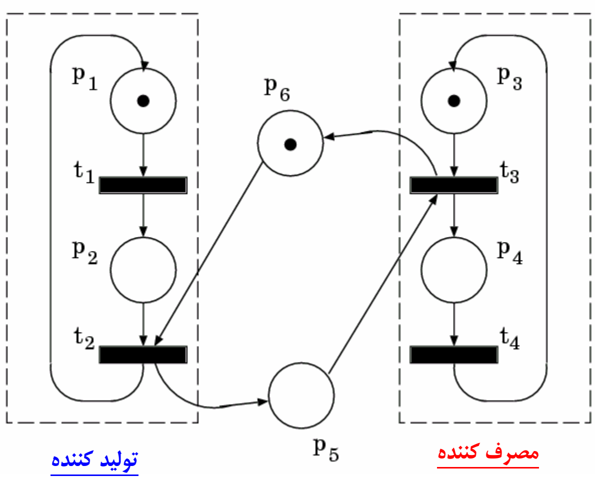
مدل‌سازی مساله تولید کننده-مصرف‌کننده

- یکی از مسایل معروف سیستم‌های همروند، مساله **تولیدکننده-مصرف‌کننده** (producer/consumer) است که در زمینه‌های مختلفی کاربرد دارد.
- ارتباط دو فرآیند تولیدکننده و مصرف‌کننده از طریق یک بافر مشترک یا یک کانال ارتباطی امکان‌پذیر است.
- نگارش‌های مختلفی از مساله تولیدکننده و مصرف‌کننده وجود دارد که ظرفیت بافر مشترک نامحدود، یک یا به تعداد مشخصی است. مدل‌های متناظر با هر کدام از این نگارشها را در ادامه می‌بینیم...

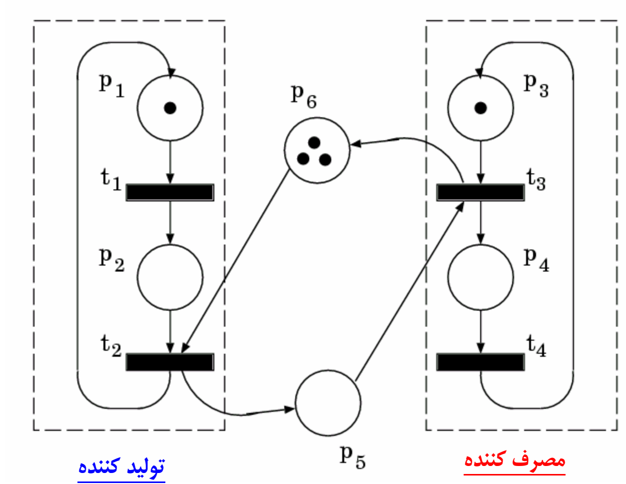
تولید کننده-مصرف‌کننده دارای بافر مشترک نامحدود



تولید کننده-مصرف‌کننده دارای بافر مشترک با ظرفیت یک



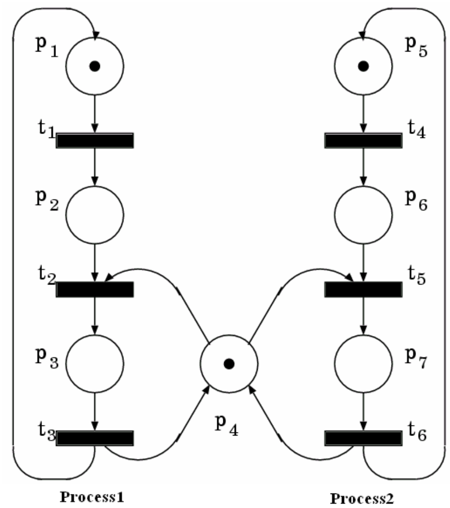
تولید کننده-مصرف‌کننده دارای بافر مشترک با ظرفیت سه



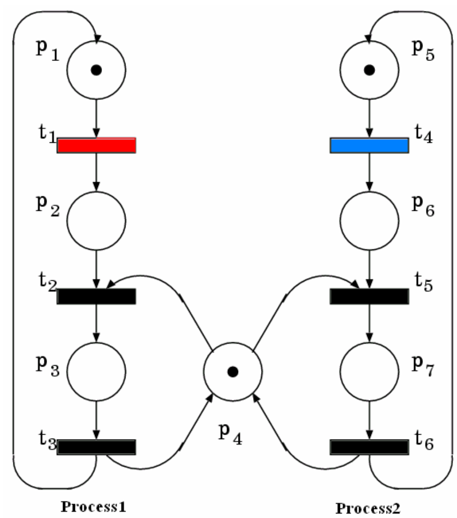
مدل‌سازی مساله مانعه‌الجمع

- یکی دیگر از مسائلی که در سیستم‌های همروند وجود دارد **مساله مانعه‌الجمع یا دو به دو انحصاری** (mutual exclusion) است.
- برای مثال یک کانال ارتباطی ممکن است که مورد دسترسی دو فرآیند قرار گیرد. اما امکان دسترسی همزمان هر دو وجود ندارد. مثلاً نوشتن و خواندن در یک فایل نمی‌تواند به‌طور همزمان انجام شود.
- در ادامه یک مدل برای این منظور ارائه می‌شود که دو فرآیند Process_1 و Process_2 به‌طور انحصاری به بافری که با مکان مشترک P_4 مدل شده است دسترسی دارند...

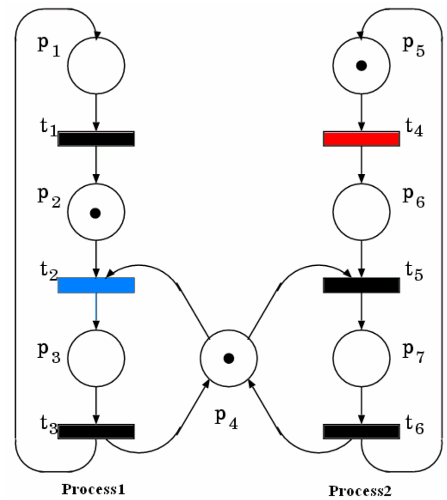
مدل‌سازی مساله مانعه‌الجمع



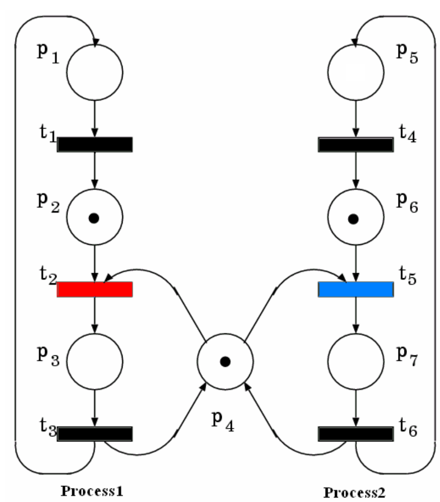
مدل‌سازی مساله مانعه‌الجمع



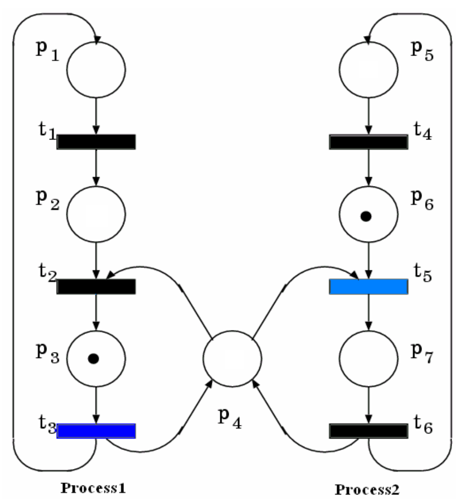
مدل‌سازی مساله مانعه‌الجمع



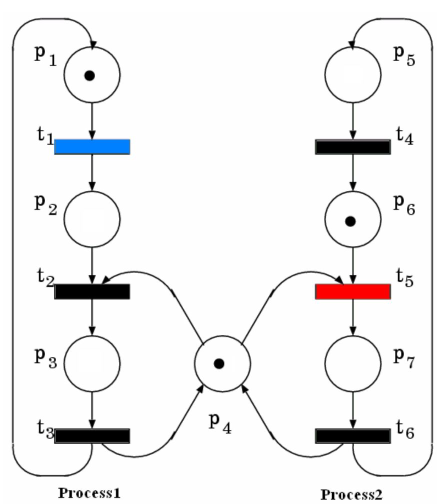
مدل‌سازی مساله مانعه‌الجمع



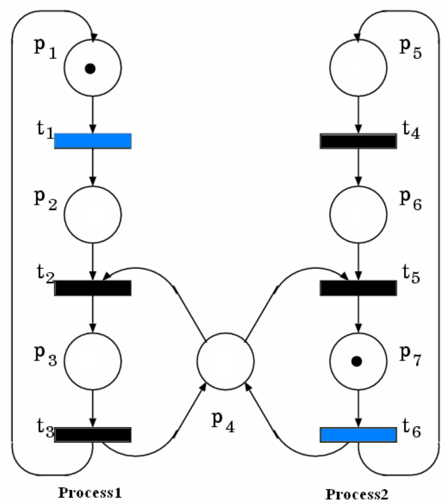
مدل‌سازی مساله مانعه‌الجمع



مدل‌سازی مساله مانعه‌الجمع



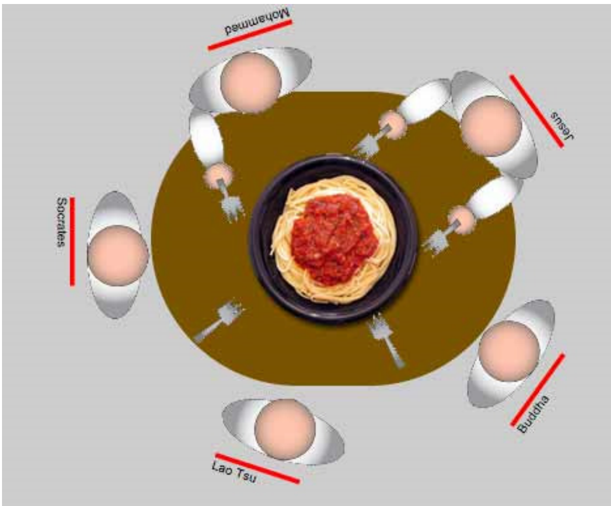
مدل‌سازی مساله مانعه‌الجمع



مساله فلاسفه در ناهارخوری

- مساله فلاسفه در ناهارخوری (dining philosophers) یک مثال کلاسیک سیستم‌های همروند است:
  - پنج فیلسوف دور یک میز نشسته‌اند، یا در حال تفکر و مباحثه هستند، یا آنکه در حال غذا خوردن هستند. هر کدام یک ظرف اسپاگتی دارند و باید دو چنگال برای خوردن آن داشته باشند. تعداد چنگالها پنج تا است که مابین فلاسفه قرار دارند...

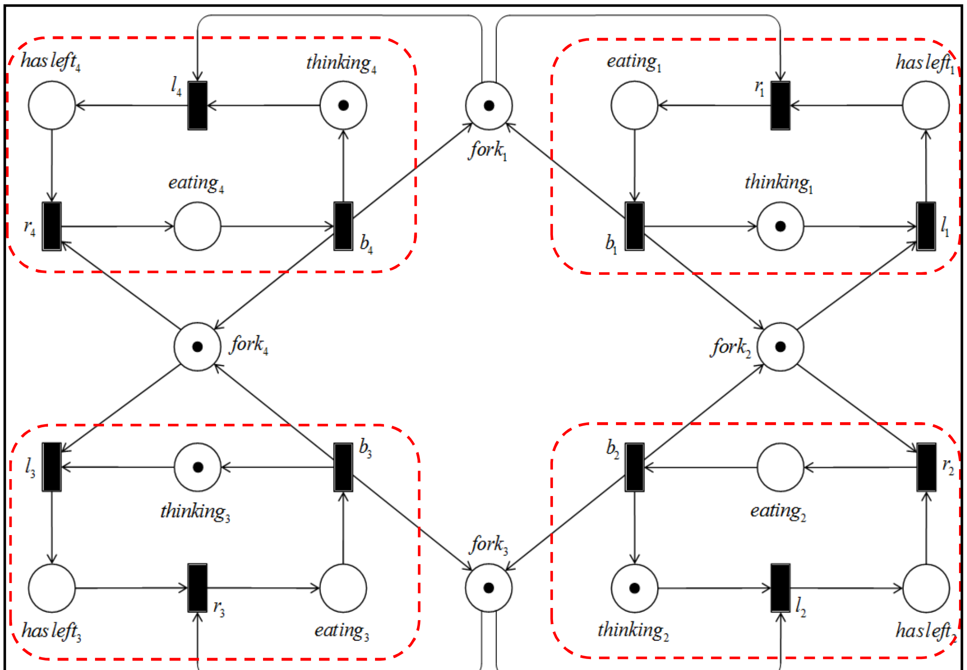
مساله فلاسفه در ناهارخوری



مساله فلاسفه در ناهارخوری

مساله فلاسفه در ناهارخوری

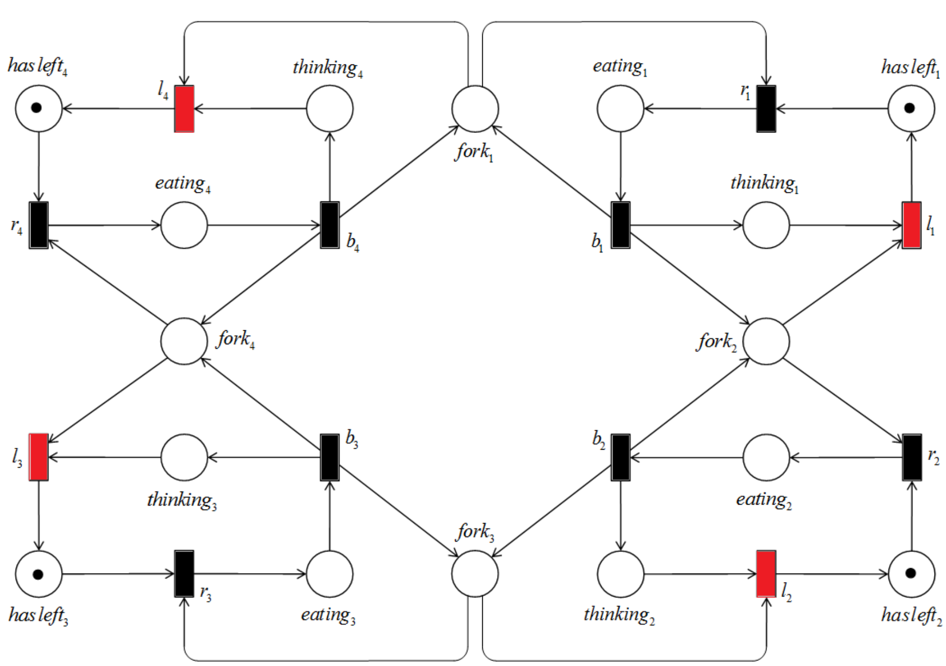
- اگر همزمان همه فلاسفه گرسنه شده و هر کدام نخست اقدام به برداشتن چنگال سمت چپ خودش کند و سپس بخواهد که چنگال سمت راست را بدست آورد، آنگاه هر کدام از آنها منتظر می‌ماند تا فیلسوف کناری چنگال برداشته شده را روی میز قرار دهد. چون او هم گرسنه است هیچ وقت این کار را نخواهد کرد. در نتیجه بن‌بست پیش خواهد آمد و همه فلاسفه از گرسنگی خواهند مرد!
- ممکن است که بخواهیم مدلی بسازیم که این مساله را به همین صورت که دارای بن بست است مدل‌سازی کند.
  - مدل برای ۴ فیلسوف است....



مدل فلاسفه در ناهارخوری دارای بن‌بست

- در این مدل اجزاء زیر را داریم:
  - مکان‌های fork1 الی fork4 برای مدل‌سازی چنگالها،
  - مکان‌های thinking1 الی thinking4 برای مدل‌سازی وضعیت در حال تفکر بودن فلاسفه،
  - گذرهای l1 الی l4 برای مدل‌سازی برداشتن چنگال سمت چپ،
  - مکان‌های hasleft1 الی hasleft4 برای مدل‌سازی وضعیت داشتن چنگال سمت چپ،
  - گذرهای r1 الی r4 برای مدل‌سازی برداشتن چنگال سمت راست،
  - مکان‌های eating1 الی eating4 برای مدل‌سازی وضعیت در حال غذا خوردن فلاسفه، و
  - گذرهای b1 الی b4 برای مدل‌سازی خاتمه غذا خوردن و آزادسازی چنگال‌ها توسط فلاسفه.
- در این مدل مکان‌های fork1 الی fork4 و مکان‌های thinking1 الی thinking4 دارای یک نشانه هستند. در نتیجه گذرهای l1 الی l4 توانا بوده و می‌توانند شلیک کنند...

FMSE 2.1 - Petri Nets - By: M. Abdollahi Azgomi - IUST-CE 95

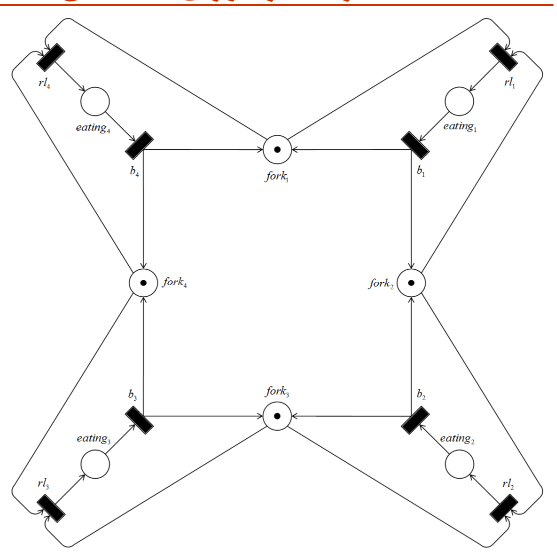


FMSE 2.1 - Petri Nets - By: M. Abdollahi Azgomi - IUST-CE 96

مدل فلاسفه در ناهارخوری دارای بن‌بست

- در این شرایط همه گذرها ناتوان بوده و در نتیجه همه آنها در شرایط بن‌بست قرار می‌گیرند.
- ممکن است بخواهیم این مشکل را بر طرف کنیم و مدلی بسازیم که فاقد بن‌بست باشد.
- برای این منظور اگر هر کدام از فلاسفه همزمان سعی در بدست آوردن هر دو چنگال کند و اگر نتواند هر دو را بدست آورد چنگال برداشته شده را بر روی میز قرار دهد مساله بن‌بست حل می‌شود. مدلی که در ادامه ارائه می‌شود چنین مدلی است...

مدل فلاسفه در ناهارخوری فاقد بن‌بست



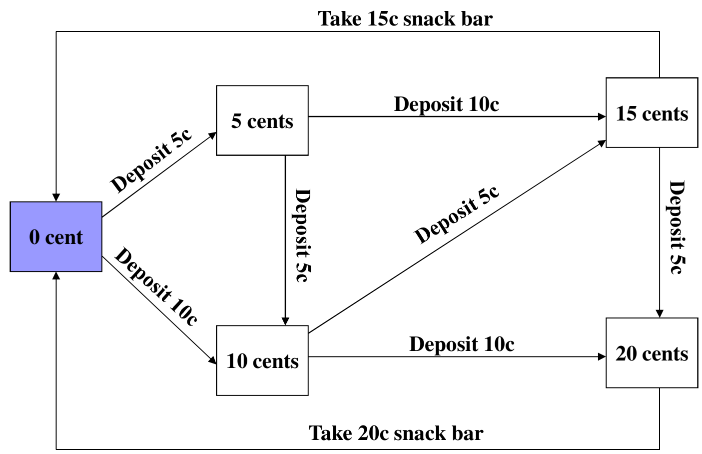
دو مثال دیگر

- در ادامه دو مثال دیگر ارائه می‌شود که در جلسات بعد ممکن است از آنها برای بحثهای دیگری استفاده کنیم:
  - مثال اول مدلی برای یک ماشین خودکار فروش خوراکی (vending machine) است.
  - مثال دوم هم مدلی برای فرآیند ورود کلمه رمز استفاده کننده از یک خودپرداز (ATM) است.

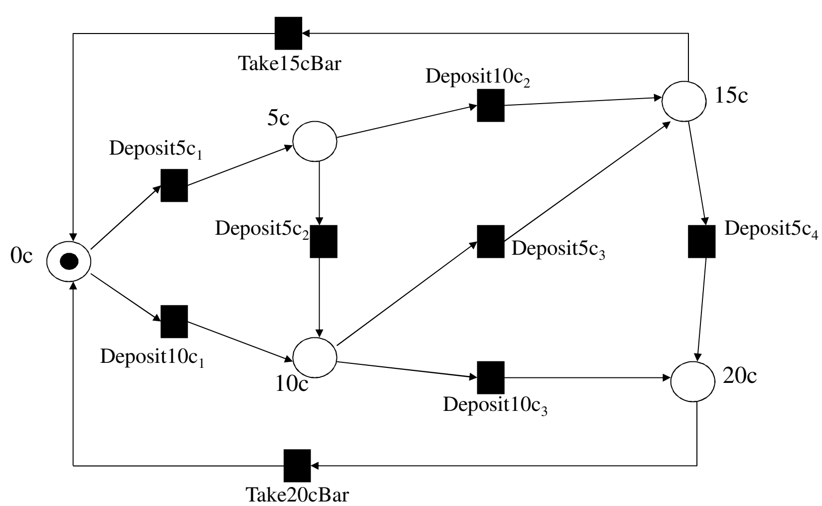
مدلی برای یک ماشین خودکار فروش خوراکی

- یک ماشین خودکار فروش خوراکی را در نظر بگیرید که دو نوع خوراکی به قیمت ۱۵ و ۲۰ سنت را می‌فروشد. این دستگاه فقط دو نوع سکه ۵ و ۱۰ سنتی را قبول می‌کند. همچنین این دستگاه پول خرد بر نمی‌گرداند. یعنی نمی‌توان سکه‌های با ارزش بیشتر را وارد نمود.
- در ادامه ابتدا مدل ماشین حالت متناهی (FSM) برای این دستگاه ارائه شده و سپس تبدیل به یک مدل شبکه پتری می‌شود...

مدل FSM برای یک ماشین خودکار فروش خوراکی



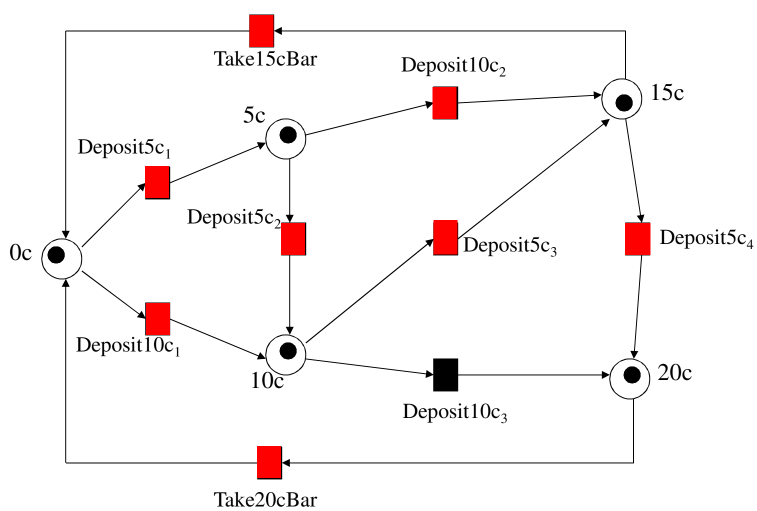
مدل شبکه پتری برای یک ماشین خودکار فروش خوراکی



مدل شبکه پتری برای یک ماشین خودکار فروش خوراکی

- سه سناریو برای اجرای این مدل متصور است:
  - سناریو ۱:
    - Deposit 5c, deposit 5c, deposit 5c, deposit 5c, take 20c snack bar.
  - سناریو ۲:
    - Deposit 10c, deposit 5c, take 15c snack bar.
  - سناریو ۳:
    - Deposit 5c, deposit 10c, deposit 5c, take 20c snack bar.
- در ادامه اجرای این سه سناریو به‌صورت انیمیشن ارائه می‌شود...

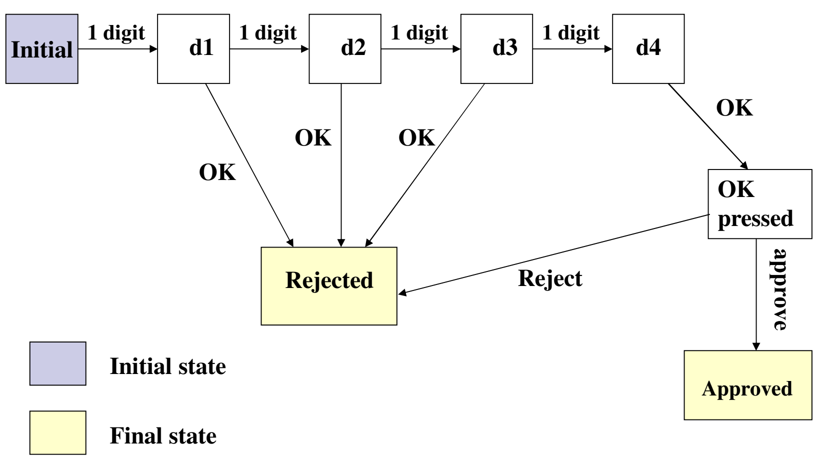
مدل شبکه پتری برای یک ماشین خودکار فروش خوراکی



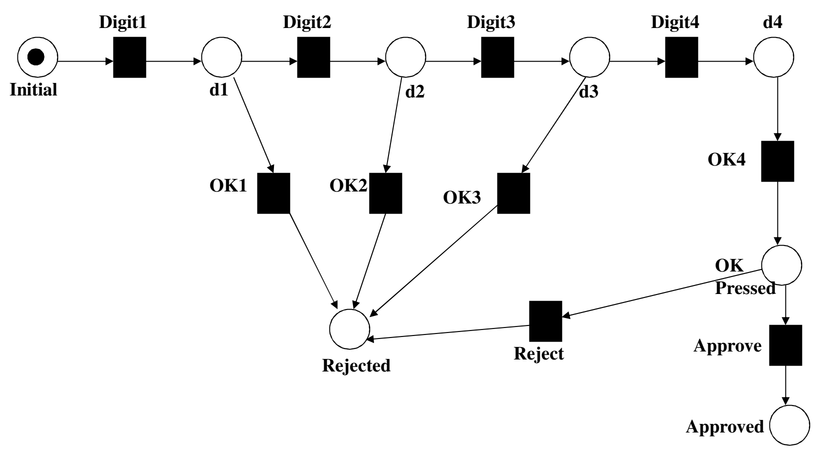
مدل‌سازی فرایند ورود کلمه رمز یک خودپرداز

- حالا مثال دیگری ارائه می‌شود که برای فرآیند ورود کلمه رمز توسط استفاده‌کننده از یک خودپرداز (ATM) است.
- کلمه رمزی که توسط استفاده کننده باید وارد شود ۴ رقمی است. پس از ورود هر رقم ممکن است استفاده‌کننده کلید OK را بزند که در این صورت اگر ۴ رقم وارد نشده باشد، طرد (reject) خواهد شد. اما پس از ورود چهارمین رقم ممکن است که کلمه رمز وارد شده درست باشد و استفاده‌کننده اجازه دسترسی به دستگاه را پیدا کند (approved). یا آنکه کلمه رمز درست نباشد و طرد شود.
- در ادامه ابتدا یک مدل FSM و سپس یک مدل شبکه پتری برای این سیستم ارائه می‌شود...

مدل FSM برای ورود کلمه رمز یک خودپرداز



مدل شبکه پتری برای ورود کلمه رمز یک خودپرداز



مدل شبکه پتری برای ورود کلمه رمز یک خودپرداز

- حالا دو سناریو را برای اجرای این مدل در نظر می‌گیریم:
  - سناریو ۱: استفاده کننده ۴ رقم را وارد نموده و کلید OK را بزنند.
  - سناریو ۲: استفاده کننده ۳ رقم را وارد نموده و کلید OK را بزنند.

مدل شبکه پتری برای ورود کلمه رمز یک خودپرداز