



عنوان درس:

## ارزیابی کارایی سیستم‌های کامپیوتری

### ۱۸- شبکه‌های پتری سطح بالا

دکتر محمد عبداللّهی آزگمی

دانشیار گروه نرم‌افزار

دانشکده مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

azgomi@iust.ac.ir

### فهرست مطالب

- مقدمه
- بسطهای مبنایی شبکه‌های پتری
- شبکه‌های پتری زمانی (timed Petri nets)
- شبکه‌های پتری تصادفی (stochastic Petri nets)
- شبکه‌های پتری تصادفی تعمیم‌یافته (generalized SPNs)
- شبکه‌های فعالیت تصادفی (stochastic activity networks)
- شبکه‌های پتری سطح بالا (high-level Petri nets)
- سایر بسطها شبکه‌های پتری

### مقدمه

- شبکه‌های پتری به دلیل آنکه تنها دارای دو مفهوم اولیه (مکان و گذر) هستند، بیش از حد ساده هستند.
- این سادگی گرچه باعث قابل فهم شدن مدل می‌شود، اما مدل‌سازی سیستم‌های پیچیده را دشوار می‌کند.
- به این دلیل مدل‌سازی با شبکه‌های پتری سهل و ممتنع است.
- در نتیجه ساخت مدل (model construction) با شبکه‌های پتری در کاربردهای واقعی بیش از آنکه علم باشد، هنر است. یعنی کسی که هنر و تجربه بالاتری دارد، به راحتی می‌تواند مدل‌سازی کند.

### مقدمه

- برای تقویت شبکه‌های پتری و فراهم‌سازی امکان مدل‌سازی سیستم‌های پیچیده، افراد مختلف شروع به بسط شبکه‌های پتری نمودند. این بسط‌ها به سه دسته قابل تقسیم هستند:
- برای افزایش قدرت مدل‌سازی (modeling power): برای مثال با کمان بازدارنده (inhibitor arc) قدرت مدل‌سازی شبکه‌های پتری معادل ماشین تورینگ (Turing machine) گردیده است. همچنین، شبکه‌های پتری زمانی (times Petri nets)، مفهوم زمان را وارد شبکه‌های پتری نموده و امکان مدل‌سازی مفهوم زمان را فراهم نموده است.
- برای تسهیل امکان تحلیل مدل‌های ایجاد شده: برای مثال شبکه‌های پتری تصادفی (stochastic Petri nets) امکان حل مدل‌های ایجاد شده را با زنجیره‌های مارکوف فراهم نموده است.
- برای فراهم‌سازی انعطاف و امکانات سطح بالا مدل‌سازی: برای نمونه شبکه‌های پتری رنگی (coloured Petri nets) امکانات زبانهای مدل‌سازی سطح بالا را به شبکه‌های پتری اضافه نمود.

### مقدمه

■ در ادامه بسط‌های معروف شبکه‌های پتری معرفی می‌شود. این بسط‌ها عبارتند از:

- بسط‌های مبنایی شبکه‌های پتری،
- شبکه‌های پتری زمانی،
- شبکه‌های پتری تصادفی،
- شبکه‌های پتری تصادفی تعمیم‌یافته،
- شبکه‌های فعالیت تصادفی، و
- شبکه‌های پتری سطح بالا.

■ تاکید بر بسط‌های تصادفی خواهد بود که در ارزیابی کارایی مورد استفاده قرار می‌گیرند.

### بسط‌های مبنایی شبکه‌های پتری

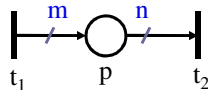
■ در ابتدا بسط‌های مبنایی شبکه‌های پتری معرفی می‌شوند، که عبارتند از:

- مضرب کمان (arc multiplicity) یا کمان چندگانه (multiple arc)،
- کمان بازدارنده (inhibitor arc)،
- سطح اولویت (priority level)، و
- تابع تواناسازی (enabling function) یا نگهبان (guard).

■ سه بسط آخر با هدف از بین بردن مشکل انفجار فضای حالت معرفی شده‌اند. چون برای جلوگیری از بی‌کران شدن فضای حالت شبکه قابل استفاده هستند.

## مضرب کمان

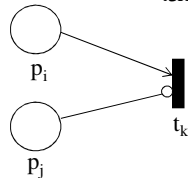
- مضرب یا عدد کمان (arc cardinality)، امکان ایجاد کمانهای چندگانه را فراهم می‌کند. این مضرب یا عدد، قابل انتساب به کمانهای ورودی و خروجی است.
- در تعریف صوری شبکه‌های پتری که در جلسه ۱۶ ارائه شد، این نوع کمان وجود داشت. ولی در تعریف اولیه شبکه‌های پتری، ارائه شده توسط خود پتری، چنین چیزی وجود نداشت.
- برای مثال مدل زیر را در نظر بگیرید:



□ وقتی گذر  $t_1$  شلیک کند  $m$  نشانه را در مکان  $p$  قرار می‌دهد و گذر  $t_2$  وقتی می‌تواند که  $n$  نشانه در مکان  $p$  قرار داشته باشد و پس از شلیک کردن  $t_2$ ،  $n$  نشانه را از مکان  $p$  برمی‌دارد.

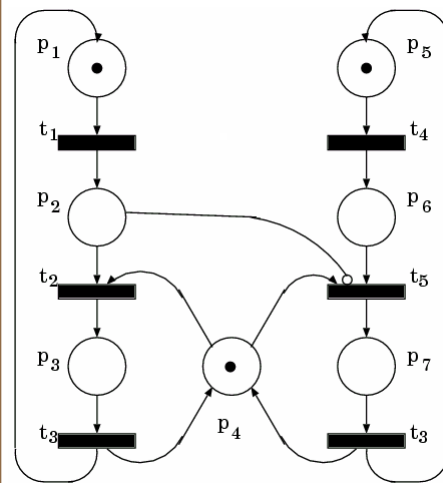
## کمان بازدارنده

- کمانهای بازدارنده (inhibitor arcs) در نمایش گرافیکی با یک دایره کوچک در انتهای کمان متصل کننده مکان به گذر مشخص می‌شوند.
- وجود کمان بازدارنده به این معنی است که گذر توانا می‌شود، اگر و فقط اگر مکان حاوی نشانه نباشد.
- برای مثال در مدل زیر گذر  $t_k$  در صورتی توانا می‌شود که در مکان  $p_i$  حداقل یک نشانه وجود داشته باشد، اما در  $p_j$  نشانه‌ای نباشد.



- کمان بازدارنده نیز در تعریف صوری شبکه‌های پتری ارائه شده در جلسه ۱۶ وجود دارد. اما در تعریف اولیه ارائه شبکه‌های پتری، وجود نداشت.
- فایده اصلی کمان بازدارنده آن است که باعث می‌شود تا قدرت مدل‌سازی شبکه پتری معادل ماشین تورینگ شود. به این ترتیب می‌توان هر الگوریتمی را که با ماشین تورینگ قابل مدل‌سازی است، با شبکه‌های پتری هم بتوان مدل‌سازی نمود.

### مثالی از کمان بازدارنده

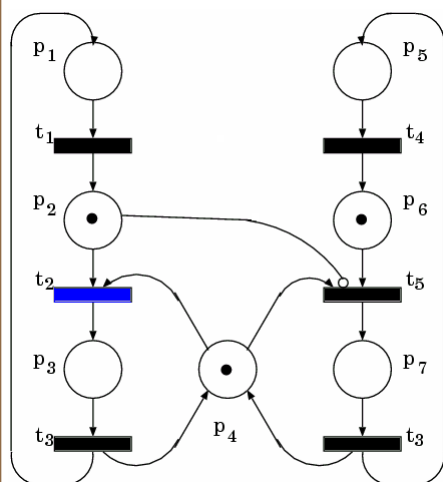


■ در شبکه پتری مقابل چه چیزی مدل سازی شده است؟

PECS 18 - High-Level Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۹

### مثالی از کمان بازدارنده



■ اولویت گذر  $t_2$  نسبت به  $t_5$  مدل سازی شده است:

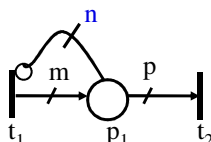
□ زیرا وقتی که  $t_1$  و  $t_4$  شلیک کنند، یک نشانه در هر کدام از مکانهای  $p_2$  و  $p_6$  قرار داده شود، در این صورت تنها گذر  $t_2$  توانا خواهد شد و به دلیل کمان بازدارنده مابین مکان  $p_2$  و گذر  $t_5$ ، این گذر توانا نخواهد شد.

PECS 18 - High-Level Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۱۰

### کمان بازدارنده چندگانه

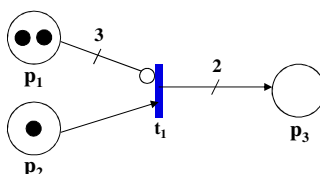
- کمان بازدارنده می‌تواند دارای یک مضرب نیز باشد. در این صورت گذر مرتبط، در صورتی که به تعدادی کمتر از مضرب در مکان متصل، نشانه وجود داشته باشد، آن گذر توانا خواهد شد.
- در شکل زیر مثالی از یک مکان بازدارنده چندگانه ارائه شده است:



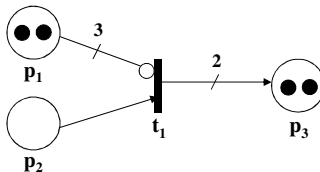
- در این مدل،  $t_1$  در صورتی توانا می‌شود که در مکان  $p_1$  کمتر از  $n$  نشانه وجود داشته باشد. در این صورت، شلیک نموده و  $m$  نشانه در مکان  $p_1$  قرار می‌دهد.
- همچنین، گذر  $t_2$  در صورتی توانا می‌شود که  $p$  نشانه در مکان  $p_1$  وجود داشته باشد.
- کمان بازدارنده چندگانه نیز در تعریف صوری شبکه‌های پتری ارائه شده در جلسه ۱۶ وجود دارد.

### مثالی از کمان بازدارنده چندگانه

- مدل زیر را ملاحظه کنید:



- در این مدل چون دو نشانه (کمتر از ۳) در مکان بازدارنده  $p_1$  قرار داشته و یک نشانه نیز در مکان ورودی  $p_2$  وجود دارد، گذر  $t_1$  توانا است.
- گذر  $t_1$  پس از شلیک کردن یک نشانه از  $p_2$  برداشته و دو نشانه در مکان خروجی  $p_3$  قرار خواهد داد:



## سطح اولویت

- **سطح اولویت (priority level)** عددی است که به گذرهای یک شبکه پتری متناسب می‌شود و تقدم آنها را نسبت به همدیگر در صورت توانا شدن همزمان برای شلیک کردن مشخص می‌کند.
- به شبکه پتری حاصله، **شبکه پتری اولویت‌دار (prioritized Petri net)** گفته می‌شود، که در آن یک گذر توانا، در صورت توانا شدن گذر دیگری که دارای اولویت بالاتر است، نخواهد توانست که شلیک نماید.
- به این ترتیب شلیک کردن گذرها از حالت غیرقطعی خارج می‌شود.
- ممکن است از هر سطح اولویت، چند گذر در مدل داشته باشیم که تشکیل یک گروه را می‌دهند.
  - مثلاً چند گذر داشته باشیم که سطح اولویت آنها ۲ بوده (گروه ۲) و چند گذر دیگر داشته باشیم که سطح اولویت آنها ۳ باشد (گروه ۳). در صورت توانا شدن همزمان، عملاً همه گذرهای گروه ۲، ناتوان خواهند بود.
  - در این صورت شلیک کردن گذرهای یک گروه دارای اولویت یکسان، به صورت غیرقطعی خواهد بود.
- لازم است که قواعد استاندارد اجرای شبکه‌های پتری برای پشتیبانی سطح اولویت اصلاح شوند.
- تنها تعداد کمی از ابزارهای شبکه‌های پتری که در پایگاه داده‌های مربوطه در نشانی وب زیر ثبت شده‌اند [1]، سطح اولویت را پشتیبانی می‌کنند:

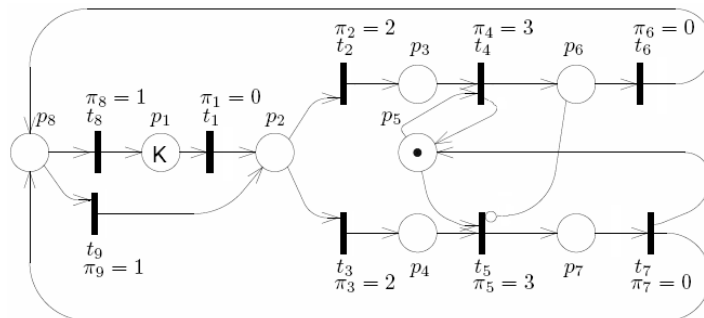
[1] [http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/complete\\_db.html](http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/complete_db.html)

## تعریف صوری شبکه پتری اولویت‌دار

- **تعریف ۱:** شبکه پتری اولویت‌دار یک نه‌تایی  $PPN = (P, T, I, O, H, \Pi, PAR, PRED, MP)$  است به نحوی که:
  - $P$ : یک مجموعه متناهی از مکانها است.
  - $T$ : یک مجموعه متناهی از گذرها است.
  - $I, O, H$ : توابع ورودی، خروجی و بازدارنده هستند.
  - $\Pi: T \rightarrow N$ : تابع اولویت (priority function) است که گذرها را به مجموعه اعداد صحیح ( $N$ ) نگاشت می‌کند. این اعداد سطح اولویت گذرها را مشخص می‌کنند.
  - $PAR$ : مجموعه‌ای از پارامترهای مدل است که برای مقداردهی اولیه مدل و تعیین تعداد نشانه‌های موجود در مکانها استفاده می‌شود.
  - $PRED$ : مجموعه‌ای از **مسئدها** (predicate) است که برای محدود کردن محدوده پارامترها استفاده می‌شود.
  - $MP: P \rightarrow N \cup PAR$ : نشانه‌گذاری اولیه است.

## مثالی از یک شبکه پتری اولویت دار

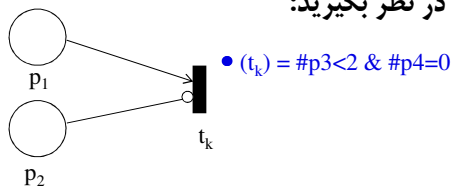
- در شکل زیر یک شبکه پتری اولویت دار نشان داده شده است، که مربوط به مساله خوانندگان و نویسندگان است.



- در شکل فوق، اولویت گذرها بالای گذرها نشان داده شده است.
- همچنین، در مدل فوق مکان  $p_1$  با استفاده از پارامتر  $k$  مقداردهی اولیه شده است. یعنی داریم:  $PAR = \{K\}$
- مسند مربوط به پارامتر فوق هم  $K \geq 1$  است. یعنی داریم:  $PRED = \{K \geq 1\}$

## تابع تواناسازی

- یک تابع تواناسازی (enabling function) که به آن نگهبان (guard) هم گفته می شود، یک عبارت بولی (Boolean expression)، متشکل از مفاهیم اولیه شبکه پتری (یعنی، مکانها، گذرها و نشانهها) است که به گذرهای یک شبکه پتری متناسب می شود.
- قاعده تواناسازی باید به نحوی اصلاح شود که علاوه بر شرایط استاندارد، تابع تواناسازی را نیز لحاظ کنند.
- برای مثال زیرمدل زیر را در نظر بگیرید:



- در این مدل گذر  $t_k$  در صورتی توانا خواهد بود که داشته باشیم:

$$\#p1 > 0 \ \& \ \#p2 = 0 \ \& \ \#p3 < 2 \ \& \ \#p4 = 0 = \text{True}$$



### شبکه‌های پتری زمانی

- زمان مفهوم مهمی در مدل‌سازی سیستم‌ها است و بدون امکان مدل‌سازی زمان، امکان ارزیابی جنبه‌های عملیاتی و وابسته به زمان سیستم‌ها وجود نخواهد داشت.
- از طرفی، شبکه‌های پتری اولیه بدون زمان (timeless) بودند.
- اما وارد نمودن زمان به شبکه‌های پتری امری بحث برانگیز بود و عده‌ای اعتقاد به آن نداشتند و ندارند.
- دلایل مخالفت با در نظر گرفتن مفاهیم زمانی (temporal concepts) در شبکه‌های پتری عبارتند از:
  - اندازه‌گیری زمان در سیستم‌های توزیع شده نیازمند همگام‌سازی (synchronization) بر مبنای یک ساعت سراسری (global) است، که مساله بحث‌انگیزی است.
  - وابستگی‌های سببی (علت و معلولی) (causal dependencies) در یک شبکه پتری بیانگر مساوی بودن زمان است.
  - در مقابل، عدم وابستگی (independency)، شکلی از توازی (parallelism) یا همروندی (concurrency) را بیان می‌کند.

### شبکه‌های پتری زمانی

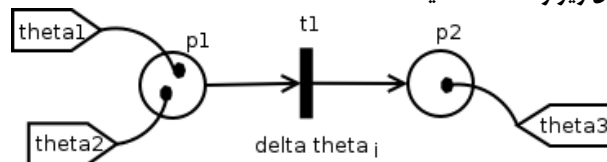
- حتی با وجودی که هنوز هم استدلال‌هایی علیه وارد نمودن زمان در شبکه‌های پتری وجود دارد، کاربردهای متعددی وجود دارند که نیازمند مفهوم زمان هستند.
- نخستین تلاش برای وارد نمودن مفهوم زمان به شبکه‌های پتری در سال ۱۹۷۴ توسط سی. رامچندانی (C. Ramchandani) در MIT انجام شد.
- از آن زمان تاکنون، رهیافت‌های متعدد و متفاوتی برای بسط دادن شبکه‌های پتری با زمان ارائه شده است.
- اما آنچه که به عنوان یک **رهیافت عمومی** (general approach) باقی مانده، انتساب زمان به گذرها به عنوان مدت **تاخیر** (delay) مابین توانا شدن و شلیک کردن یا کامل شدن آن است.
  - مدت زمان این تاخیر می‌تواند قطعی (deterministic) (ثابت و از قبل تعیین شده) یا تصادفی (stochastic) (بر اساس یک مدل احتمالی) باشد.

## رهیافت‌های شبکه‌های پتری زمانی

- رهیافت‌های مختلفی برای شبکه‌های پتری زمانی معرفی شده‌اند که زمان را به یکی از اجزاء شبکه پتری متناسب می‌کنند:
  - ☐ انتساب زمان به نشانه‌ها،
  - ☐ انتساب زمان به کمانها،
  - ☐ انتساب زمان به مکانها، و
  - ☐ انتساب زمان به گذرها (رهیافت عمومی).
- در ادامه رهیافت‌های فوق پتری معرفی می‌شوند.

## انتساب زمان به نشانه‌ها

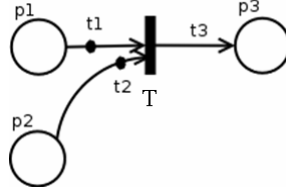
- در این رهیافت، به هر نشانه یک مهر زمانی (time-stamp) که با  $\theta$  نشان داده می‌شود، متناسب می‌شود.
- این مهر زمانی مشخص می‌کند که نشانه چه موقع برای شلیک کردن قابل دسترس است.
- برای مثال مدل زیر را مشاهده کنید:



- ☐ دو نشانه در مکان  $p1$  وجود دارند. نشانه بالایی وقتی قابل دسترس برای  $t1$  است که ساعت سراسری برابر با  $\theta_{theta1}$  شده و نشانه پایینی موقعی که ساعت برابر با  $\theta_{theta2}$  بشود.
- ☐ گذر  $t1$  پس از شلیک کردن یک نشانه را از مکان  $p1$  برداشته و مهر زمانی آن را به اندازه  $\delta\theta_i$  افزایش داده و در مکان خروجی  $p2$  قرار می‌دهد.
- این نگرش در شبکه‌های پتری رنگی (coloured Petri nets) مورد استفاده قرار گرفته است.
- عیب این رهیافت آن است که برای ارزیابی کارایی مناسب نبوده و مدل‌های حاصله تنها امکان شبیه‌سازی را خواهند داشت.

### انتساب زمان به کمانها

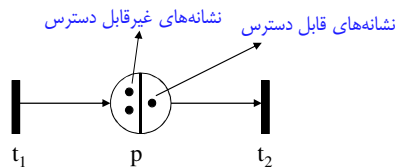
- در این رهیافت، به هر کمان یک **تاخیر سیر** (traveling delay) ( $t$ ) منتسب می‌شود.
- نشانه‌ها هنگام برداشته شدن از مکان ورودی، پس از مدت تاخیر سیر به گذر رسیده و نشانه‌هایی که قرار است در یک مکان خروجی گذاشته شوند نیز پس از مدت تاخیر سیر مربوطه، داخل آن مکان قرار می‌گیرند.
- برای مثال مدل زیر را در نظر بگیرید:



- در مدل فوق برداشته شدن نشانه‌ها از مکانهای ورودی  $p1$  و  $p2$  پس از مدت زمان  $\max(t1, t2)$  تکمیل شده و سپس بعد از مدت زمان  $t3$  یک نشانه‌ها در مکان خروجی  $p3$  گذاشته می‌شود. در نتیجه کل زمان شلیک کردن عبارت خواهد بود از:  $\max(t1, t2) + t3$
- این رهیافت خیلی مرسوم نیست.

### انتساب زمان به مکانها

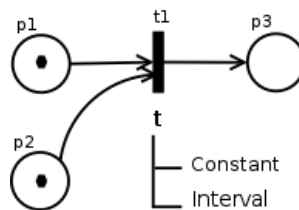
- به هر مکان ( $p$ ) یک **خاصیت تاخیر** (delay attribute) ( $t$ ) منتسب می‌شود.
- نشانه‌های قرار داده شده در یک مکان پس از سپری شدن مدت زمانی به اندازه  $t$  قابل دسترس توسط گذر متصل به آن مکان خواهند بود.
- برای مثال مدل زیر را در نظر بگیرید:



- در این مدل، دو تا از نشانه‌های موجود در  $p$ ، که مدت زمان  $t$  از زمان قرار داده شدن آنها در  $p$  سپری نشده، هنوز قابل دسترس توسط  $t2$  نبوده و یکی از آنها که مدت زمان  $t$  از قرار داده شدن در  $p$  سپری شدن، قابل دسترس است.
- این رهیافت نیز خیلی مرسوم نیست.

### انتساب زمان به گذرها

- در این رهیافت که رهیافت عمومی و اصلی است، زمان به گذرها منتسب می‌شود.
- بر این اساس، هر گذر یک **فعالیت** (activity) را مدل‌سازی می‌کند:
  - ☐ توانا شدن گذر به منزله شروع فعالیت است.
  - ☐ شلیک کردن گذر به منزله خاتمه فعالیت است.
- زمان منتسب شده به گذرها ممکن است که ثابت بوده یا یک بازه باشد:



PECS 18 - High-Level Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۲۳

### انتساب زمان به گذرها

- دو رهیافت برای انتساب زمان به گذرها معرفی شده است:
  - ☐ رهیافت رامچندنی (Ramchandani's approach)
  - ☐ رهیافت مرلین (Merlin's approach)

PECS 18 - High-Level Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۲۴

### رهیافت رامچندنی

- توسط رامچندنی در سال ۱۹۷۴ معرفی شده است [1].
- در این روش یک مدت زمان شلیک کردن (firing duration)  $(t)$  به هر گذر شبکه پتری متناسب می‌شود.
- قاعده شلیک کردن هم به صورت زیر اصلاح می‌شود:
  - گذرها به محض آنکه توانا می‌شوند شروع به شلیک کردن می‌کنند.
  - شلیک کردن پس از مدت زمان  $(t)$  کامل می‌شود.

[1] Ramchandani, C.: *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*, Ph.D. Thesis, MIT, Department of Electrical Engineering (1974)

### ارزیابی کارایی با شبکه‌های پتری زمانی

- روش رامچندنی در ارزیابی کارایی دارای کاربرد است.
- افراد دیگری غیر از رامچندنی نیز بسط‌های زمانی خاص خود را برای کاربرد در ارزیابی کارایی تعریف نموده‌اند. از جمله این افراد زوبرک (W. M. Zuberek) است [1].
- زوبرک استاد دانشگاه مموریال (Memorial) کانادا است و هم اکنون هم در این زمینه فعال است.



- مقاله زوبرک در زمینه کاربرد شبکه‌های پتری زمانی در ارزیابی کارایی جزء مراجع مهم بوده و ارجاعات فراوانی به آن انجام شده است.

[1] Zuberek, W.M.: *Timed Petri Nets and Preliminary Performance Evaluation*, Proc. 7<sup>th</sup> Annual Symposium on Computer Architecture, La Baule, France, pp. 89-96 (1980)

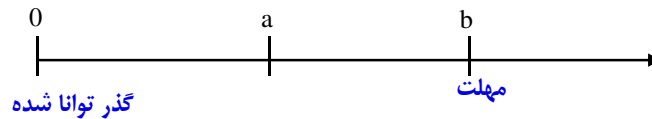
## رهیافت مرلین

■ این رویافت توسط مرلین (P. M. Merlin) در سال ۱۹۷۴ معرفی شده است [1].

■ در این روش دو عدد  $a$  و  $b$  به هر گذر شبکه پتری منتسب می‌شود به نحوی که:  $0 \leq a \leq b \leq \infty$

- $a$ : مدت زمانی را مشخص می‌کند که باید بین توانا شدن تا شروع شلیک کردن گذر باید سپری شود.
- $b$ : حداکثر زمانی است که در طی آن گذر می‌تواند توانا باقی بماند بدون آنکه شلیک کند. در حقیقت مدت زمان  $b$ ، مهلت (deadline) گذر محسوب می‌شود.

■ مطالب فوق در شکل زیر نشان داده شده‌اند:



[1] P. M. Merlin and D. J. Farber, "Recoverability of Communication Protocols: Implications of a Theoretical Study," *IEEE Transactions on Communications*, 24(9), pp.1036–1043 (1976)

## رهیافت مرلین

■ روش مرلین برای مدل‌سازی گردش کار (workflow modeling) مناسب است.

■ کارهای فان هی (K. van Hee) فان در آلت (W. van der Aalst) در همین زمینه است، که هر دو هم اکنون استاد دانشگاه آیندهون هلند هستند.



□ بسط‌هایی با نام شبکه‌های گردش کار (workflow nets) و YAWL (yet another workflow language) در اصل مبتنی بر روش مرلین هستند.

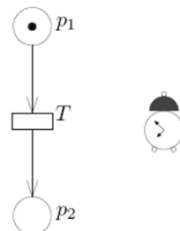
### شبکه‌های پتری زمانی

- در اینجا ما شبکه‌های پتری زمانی را که مبتنی بر رهیافت عمومی بوده و به آن صورتی که در [1] معرفی شده و بعداً هم برای تعریف بسط‌های تصادفی شبکه‌های پتری از آن استفاده خواهیم کرد، ارائه می‌کنیم.
- تعریف شبکه‌های پتری زمانی به نحوی ارائه می‌شود که رفتار مدل‌های پتری بدون زمان (یعنی با حذف پارامترهای زمانی گذرها)، بدون تغییر باقی بماند. دلیل این امر آن است که هنوز بتوان مثل سابق از شبکه‌های پتری در مطالعات تحلیل رفتاری در کنار تحلیل کارایی استفاده نمود.

[1] Ajmone Marsan, M., Balbo, G., Conte, G., Donatelli, S. and Franceschinis, G.: *Modelling with Generalized Stochastic Petri Nets*, John Wiley & Sons (1995)

### شبکه‌های پتری زمانی

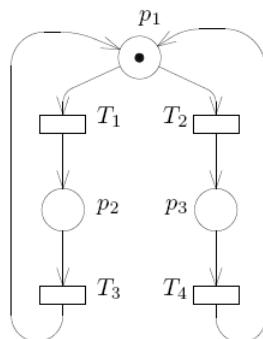
- در شبکه‌های پتری زمانی برای مدل‌سازی رخدادهایی که اجرای آنها برای یک مدت زمان مشخصی طول می‌کشد، از **گذر زمانی** (timed transition) استفاده می‌شود.
- در نمایش گرافیکی، گذرهای زمانی با مستطیل‌های توخالی نشان داده می‌شوند:



□ در این شکل وقتی یک نشانه در مکان  $p_1$  قرار گیرد، گذر  $T$  توانا شده و بلافاصله یک زمان‌سنج (timer) به مقداری مثل  $\theta$ ، مقداردهی اولیه می‌شود. این زمان‌سنج، شروع به کاهش می‌کند تا صفر شود. در این صورت، گذر شلیک نموده و یک نشانه از مکان  $p_1$  برداشته شده و یک نشانه به مکان  $p_2$  اضافه می‌شود.

## مثالی از یک شبکه پتری زمانی

■ در شکل زیر یک شبکه پتری زمانی ارائه شده است:



□ مدل فوق دارای سه مکان  $p_1$  و  $p_2$  و  $p_3$  و گذرهای زمانی  $T_1$ ،  $T_2$ ،  $T_3$  و  $T_4$ ، که هر کدام به ترتیب دارای پارامترهای زمانی  $\theta_1$ ،  $\theta_2$ ،  $\theta_3$  و  $\theta_4$  هستند.

□ نشانه گذاری اولیه مدل نیز در شکل فوق آمده است:  $M_1 = (1, 0, 0)$

## تحلیل رفتاری مثال شبکه پتری زمانی

■ در نشانه گذاری اولیه مدل، هم گذر  $T_1$  و هم  $T_2$  توانا هستند. اینکه کدام یک شلیک می کنند، با پارامترهای زمانی آنها مشخص می شود:

□ اگر  $\theta_2 < \theta_1$ ، آنگاه گذر  $T_2$  شلیک می کند،

□ و گر نه گذر  $T_2$  شلیک خواهد نمود.

■ فرض می کنیم که  $\theta_1 < \theta_2$  باشد:

□ آنگاه گذر زمان سنج های هر دو گذر مقداردهی شده و شروع به کسر شدن می کنند.

□ زمان سنج  $T_1$  زودتر صفر شده و این گذر شلیک می کند.

□ در این صورت، یک نشانه از مکان  $p_1$  برداشته شده و یک نشانه در مکان  $p_2$  گذاشته می شود و نشانه گذاری جدید  $M_2 = (0, 1, 0)$  ظاهر می شود.

□ در نتیجه، گذر  $T_2$  ناتوان (disable) می شود.



### تحلیل رفتاری مثال شبکه پتری زمانی

- در نشانه‌گذاری  $M_2$ ، گذر  $T_3$  توانا می‌شود و پس از مدت زمان  $\theta_3$ ، شلیک می‌کند. در این صورت دوباره همان نشانه‌گذاری اولیه  $M_1$  ظاهر می‌شود.
- در نتیجه، دوباره گذرهای  $T_1$  و  $T_2$  توانا می‌شوند. در این شرایط، اینکه زمان سنج دو گذر چگونه دوباره مقداردهی اولیه می‌شوند وابسته به سیستم مدل‌سازی شده است:
  - ممکن است هر دو زمان سنج دوباره مقداردهی اولیه شود. در این صورت، هیچ وقت گذر  $T_2$  شلیک نخواهد کرد.
  - اما ممکن است که زمان سنج  $T_1$  به  $\theta_1$  مقداردهی اولیه شود، ولی زمان سنج گذر  $T_2$  مقداردهی اولیه نشده و از مقدار باقیمانده  $\theta_1 - \theta_2$  شروع به کسر شدن کند و در نتیجه ممکن است که این دفعه  $T_2$  شلیک کند.

### ارزیابی شبکه‌های پتری زمانی

- شبکه‌های پتری زمانی که دارای پارامترهای زمانی عمومی باشند، دارای مشکل ارزیابی با روشهای حل عددی هستند. در نتیجه کاربرد آنها در ارزیابی کارایی با مشکلاتی مواجه است.
- گرچه تلاشهایی برای یافتن روشهای تحلیلی شبکه‌های پتری زمانی شده است، اما راه حل اصلی برای ارزیابی آنها، همانا شبیه‌سازی گسسته-رخداد است.
- برای سهولت ارزیابی شبکه‌های پتری زمانی، شبکه پتری تصادفی (SPN: stochastic Petri nets) معرفی شده است که در ادامه با آن آشنا خواهیم شد.

## شبکه‌های پتری تصادفی

- همانگونه که گفتیم با انتساب زمان به شلیک کردن گذرها، شبکه‌های پتری زمانی حاصل می‌شوند.
  - یک حالت خاص شبکه‌های پتری زمانی، شبکه پتری تصادفی (SPN: stochastic Petri nets) است که زمانهای شلیک کردن، متغیرهای تصادفی در نظر گرفته می‌شوند.
  - یک حالت خاص شبکه‌های پتری تصادفی آن است که زمانهای شلیک کردن دارای توزیع نمایی باشد.
  - ملوی (M. K. Molloy) در سال ۱۹۸۱ با طرح این ایده نشان داد که فرآیند نشانه‌گذاری (marking process) یا فرآیند حالت (state process) شبکه پتری تصادفی به یک CTMC که فضای حالت آن همشکل یا همریخت (isomorphic) با گراف دسترس‌پذیری شبکه پتری باشد، قابل نگاشت است [1].
  - این نتیجه بسیار مهمی است. چون امکان ارزیابی شبکه‌های پتری تصادفی را با تحلیل CTMC فراهم می‌کند.
- [1] Molloy, M.K.: "Performance Analysis Using Stochastic Petri Nets", *IEEE Trans. Computers* 31(9), pp. 913-917 (1982)

## تبدیل SPNs به CTMC

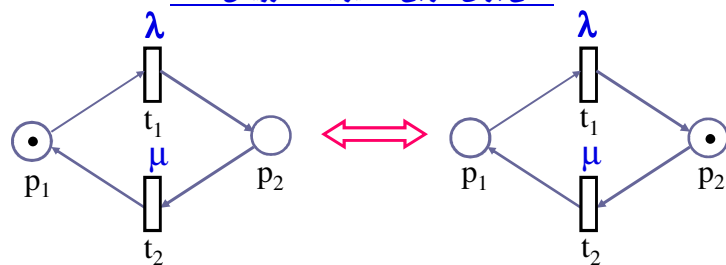
- برای حل مدل‌های SPNs مراحل زیر را انجام می‌دهیم:
  1. گراف دسترسی مدل را بدست می‌آوریم:
    - برای این منظور یک الگوریتم معمولی تولید گراف دسترسی شبکه‌های پتری را می‌توانیم استفاده کنیم.
  2. در گراف دسترسی حاصله:
    - برچسب کمانها را از نام گذر زمانی به نرخ آن گذر نمایی تبدیل نموده و
    - نام گره را از بردار نشانه‌گذاری به یک شماره یکتا تبدیل می‌کنیم.
- آنچه بدست خواهد آمد نمایش یک CTMC است.
- در نتیجه، مشخص می‌شود که استفاده از یک مدل سطح بالایی مثل SPNs ابزاری برای تولید خودکار CTMC است. به این دلیل با استفاده از این مدل‌ها، روشهای تولید CTMC (CTMC generation methods) گفته می‌شود. در نتیجه، نیازی به شناسایی حالت‌های زیاد یک مدل واقعی و بدست آوردن دستی CTMC نخواهیم بود.

### تبدیل SPNs به CTMC

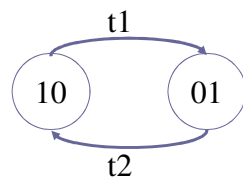
- حالا می‌توانیم طبق روشهای حل حالت پایدار یا حل گذرا که قبلاً برای CTMC معرفی نمودیم، عمل کنیم و مدل SPN را ارزیابی کنیم.
- تنها نکته‌ای که باقی می‌ماند، نحوه محاسبه معیارهای کارایی در سطح مفاهیم اولیه SPNs است که بعداً به آن اشاره خواهیم کرد.
- در ادامه مثالهایی را برای تشریح نحوه بدست آوردن CTMC از SPNs خواهیم دید....

### مثال اول تبدیل SPN به CTMC

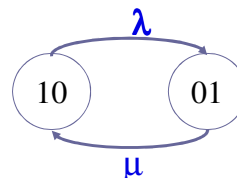
مدلی برای خرابی/تعمیر یک سرویس‌دهنده



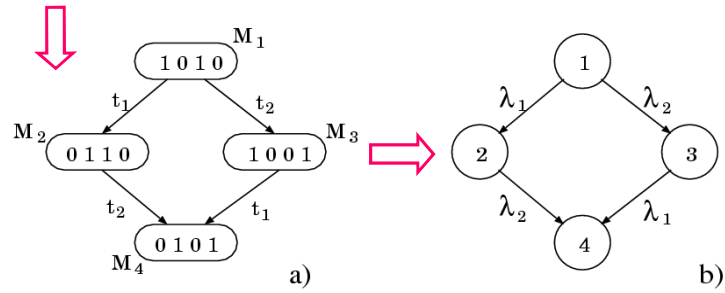
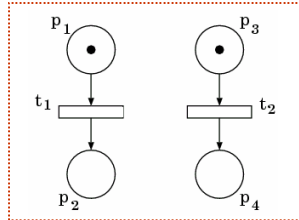
گراف دسترس پذیری



CTMC



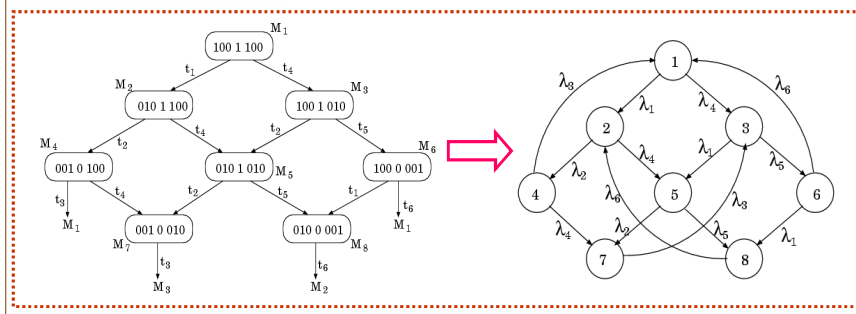
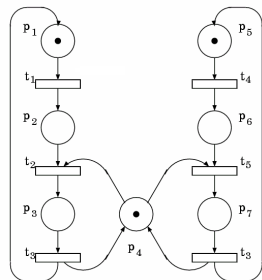
### مثال دوم تبدیل SPN به CTMC



PECS 18 - Analysis of Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۳۹

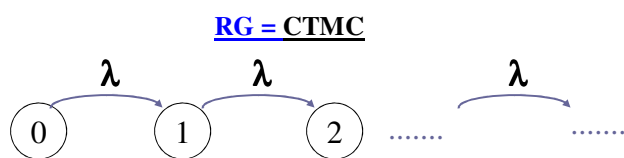
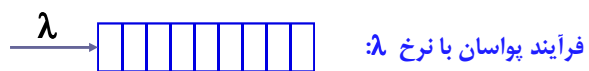
### مثال سوم تبدیل SPN به CTMC



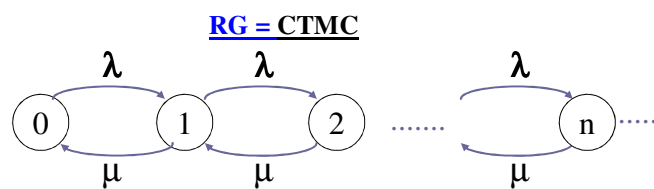
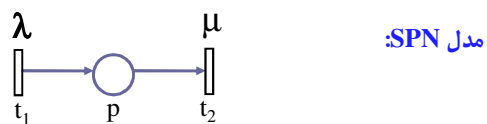
PECS 18 - Analysis of Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۴۰

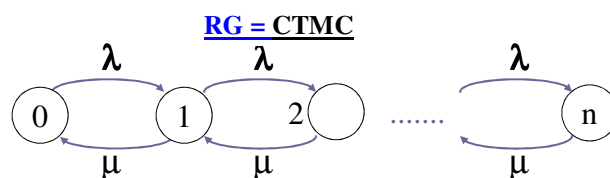
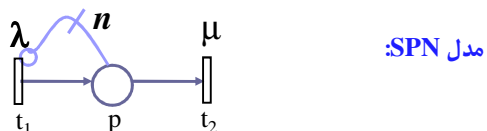
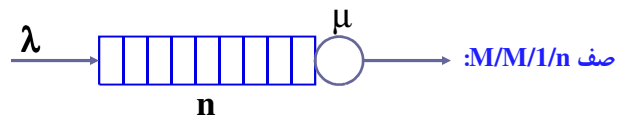
### مدل سازی فرآیند پواسان با SPN



### مدل سازی صف M/M/1 با SPN



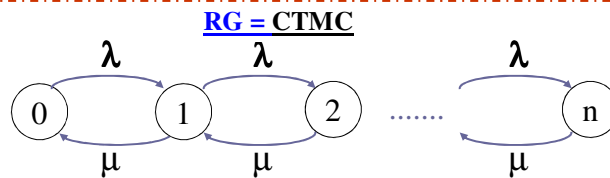
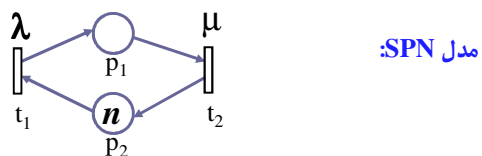
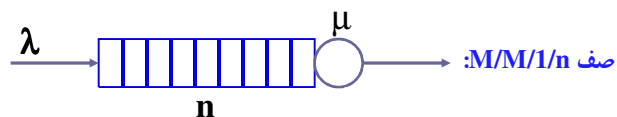
### مدل سازی صف M/M/1/n با SPN (نگارش ۱)



PECS 18 - Analysis of Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۴۳

### مدل سازی صف M/M/1/n با SPN (نگارش ۲)

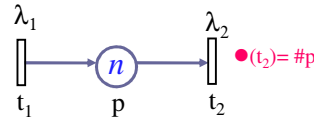


PECS 18 - Analysis of Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۴۴

### نرخ شلیک کردن وابسته به حالت

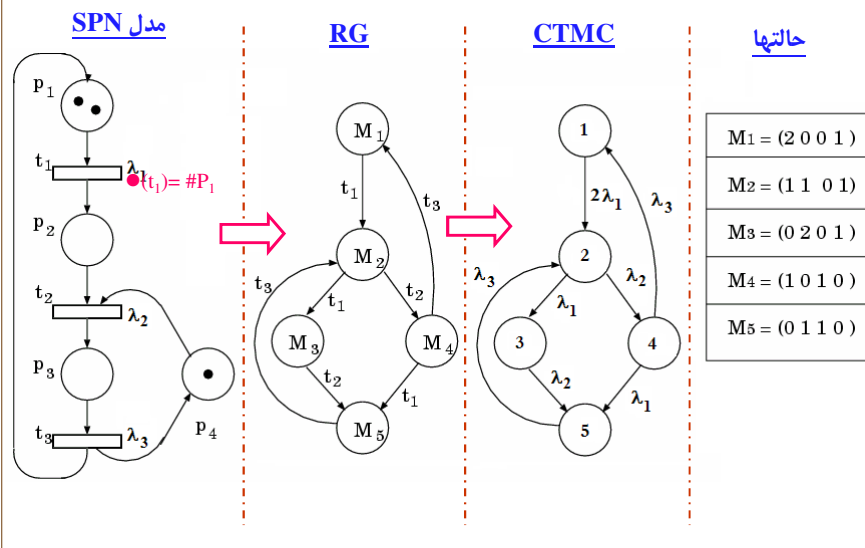
- در SPN به هر گذر زمانی یک نرخ شلیک کردن (firing rate) متناسب می شود. این نرخ می تواند وابسته به نشانه گذاری (marking dependent) باشد.
- برای مثال در شکل زیر، نرخ گذر  $t_2$  به تعداد نشانه های مکان  $p$  وابسته است:



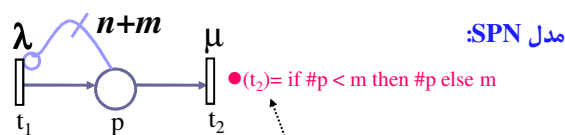
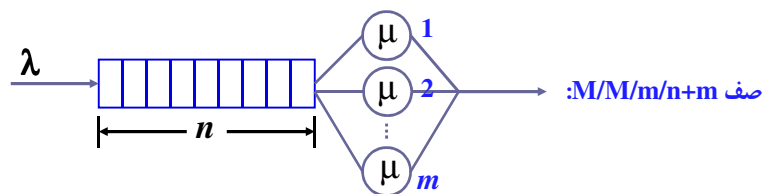
□ با توجه به وابسته بودن نرخ گذر  $t_2$  به تعداد نشانه های مکان  $p$ :

- اگر هیچ نشانه ای در مکان  $p$  نداشته باشیم، طبیعتاً نرخ  $t_2$  صفر است (در حقیقت اصلاً توانا نیست).
- اگر تعداد نشانه های  $p$  یک باشد، نرخ  $t_2$  برابر با  $\lambda_2$  خواهد بود.
- اگر تعداد نشانه های  $p$  دو باشد، نرخ  $t_2$  برابر با  $2\lambda_2$  خواهد بود.
- در حالت کلی نرخ  $t_2$  برابر با  $n\lambda_2$  خواهد بود که  $n$  تعداد نشانه های مکان  $p$  است.

### مثال ۱: مدل سازی مانع الجمع با SPN



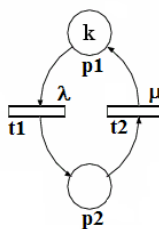
## مثال ۲: مدل سازی صف M/M/m/n+m با SPN



نرخ شلیک کردن وابسته به حالت

## مثال ۳: مدل سازی سیستمی که دارای k قطعه قابل تعمیر است

- مدل SPN برای سیستمی که دارای k قطعه قابل تعمیر (repairable components) است به شکل زیر است:



- در این مدل:

- گذر  $t_1$  خراب شدن قطعات با نرخ  $\lambda$  را مدل سازی نموده و  $t_2$  تعمیر شدن قطعات خراب با نرخ  $\mu$  را مدل سازی می کند.
- $p_1$  و  $p_2$  هم به ترتیب تعداد قطعات سالم و خراب را نشان می دهند، که  $p_1$  با پارامتر  $k$  مقداردهی اولیه شده است.
- با توجه به اینکه تعداد تعمیرکاران ممکن است یک یا دو باشد، در ادامه نرخ گذرهای  $t_1$  و  $t_2$  را تعریف می کنیم...

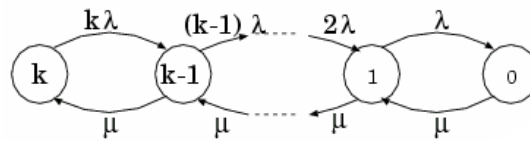


### مثال ۳: مدل سازی سیستمی که دارای $k$ قطعه قابل تعمیر است

■ اگر تعداد تعمیرکاران یک باشد، نرخ گذرها به صورت زیر خواهد بود:

$$\begin{aligned} \bullet(t_1) &= \#P_1 \\ \bullet(t_2) &= \end{aligned}$$

■ در این صورت CTMC متناظر با این مدل به صورت زیر خواهد بود:

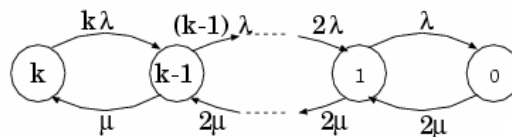


### مثال ۳: مدل سازی سیستمی که دارای $k$ قطعه قابل تعمیر است

■ اما اگر تعداد تعمیرکاران دو باشد، نرخ گذرها به صورت زیر خواهد بود:

$$\begin{aligned} \bullet(t_1) &= \#P_1 \\ \bullet(t_2) &= \begin{cases} \#P_2 & \text{if } \#P_2 < 2 \\ 2 & \text{otherwise} \end{cases} \end{aligned}$$

■ در این صورت CTMC متناظر با این مدل به صورت زیر خواهد بود:



### شبکه‌های پتری تصادفی تعمیم یافته

- با معرفی شبکه‌های پتری زمانی و تصادفی، امکان مدل‌سازی عدم قطعیت، که یکی از نیازهای اساسی برای مدل‌سازی سیستم‌های همروند است، به آن صورت که در تعریف اولیه شبکه‌های پتری وجود داشت از بین رفت:
  - زیرا تصمیم‌گیری در مورد اینکه کدام گذر از میان گذرهای زمانی توانا شلیک می‌کند، بر اساس زمان شلیک کردن آنها انجام می‌شود، نه اینکه به‌طور غیرقطعی یکی از آنها برای شلیک کردن انتخاب شود.
- علاوه بر آن، خیلی از اوقات در سیستمی که قرار است مدل‌سازی شود، برخی از کنشها، نظیر تصمیم‌گیری در مورد انجام یکی از دو کنش امکان‌پذیر بر اساس شرایط مربوطه و مشابه جمله *if-else* در زبانهای برنامه‌سازی، به‌طور آنی (instantaneous) انجام می‌شوند.
- یعنی کنشهایی داریم که در مقایسه با کنشهای اصلی سیستم، زمان اجرای آنها آنقدر ناچیز است که می‌توانیم آن را مساوی صفر فرض کنیم.

### شبکه‌های پتری تصادفی تعمیم یافته

- یکی از کسانی که در فاصله سالهای ۱۹۸۰-۱۹۸۱ در گروه تحقیقاتی ملوی بر روی شبکه‌های پتری تصادفی کار می‌کرد، دانشمند ایتالیایی مارکو آجمان مرسان (M. Ajmone Marsan) بود.
- به دلیل اهمیت مدل‌سازی عدم قطعیت، اجمان مرسان ایده افزودن یک نوع گذر جدید را به شبکه‌های پتری تصادفی مطرح نمود که معادل گذرهای اولیه شبکه پتری بوده و زمان اجرای آن صفر بوده و به کمک آن می‌توان عدم قطعیت را مدل‌سازی نمود.
- اما ملوی با این ایده مخالفت نمود. دلیل این امر آن بود که ملوی استدلال می‌کرد که با افزودن این نوع گذر جدید، دیگر فرآیند حالت شبکه‌های پتری تصادفی حاصله، با CTMC هم‌ریخت نخواهد بود و نمی‌توان مدل‌های حاصله را حل نمود.



## شبکه‌های پتری تصادفی تعمیم یافته

- اجمان مرسان پس از مراجعت به ایتالیا و دانشگاه پلی تکنیک تورینو، ایده خود را عملیاتی نمود و در سال ۱۹۸۴ شبکه‌های پتری تصادفی تعمیم یافته (GSPN: generalized stochastic Petri net) را معرفی نمود [1].
- امروزه GSPNs پرکاربردترین بسط تصادفی شبکه‌های پتری بوده که در کاربرد ارزیابی کارایی مورد استفاده قرار می‌گیرد و توسط ابزارهای مدل‌سازی پرقدرتی پشتیبانی می‌شود.
- در ادامه مدل‌های GSPNs و روشهای حل آنها را معرفی می‌کنیم.

[1] Ajmone Marsan, M., Balbo, G., and Conte, G.: "A Class of Generalized Stochastic Petri Nets for the Performance Analysis of Multiprocessor Systems," *ACM Transactions on Computer Systems* 2(1) (1984)

## شبکه‌های پتری تصادفی تعمیم یافته

### ■ در مدل‌های GSPNs دو نوع گذر وجود دارد:

#### □ گذرهای زمانی (timed transition):

- معادل گذرهای SPNs است که زمان شلیک کردن گذرهای توانا شده، از یک توزیع نمایی بدست می‌آید.
- در نمایش گرافیکی، گذرهای زمانی با یک مستطیل توخالی داده می‌شوند:
- برای هر گذر زمانی یک نرخ برای توزیع زمان شلیک کردن گذر مشخص می‌شود.

$\lambda$

#### □ گذرهای فوری (immediate transition):

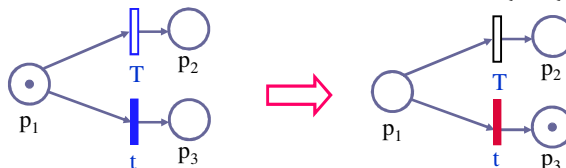
- گذرهای جدیدی هستند که زمان شلیک کردن آنها برابر صفر است.
- در نمایش گرافیکی، گذرهای فوری با یک مستطیل توپر نشان داده می‌شوند:
- قوانین اتصال این نوع گذر به مکانها مشابه گذرهای زمانی است.
- برای هر گذر فوری، یک احتمال به عنوان **احتمال شلیک کردن** (firing probability) مشخص می‌شود که در مورد آن در ادامه صحبت خواهیم نمود.

$p$

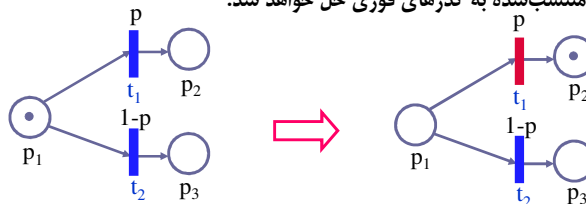
## شبکه‌های پتری تصادفی تعمیم یافته

■ قواعد تواناسازی متناسب با افزوده شدن گذرهای فوری اصلاح می‌شوند:

□ اگر در یک نشانه‌گذاری هم یک گذر زمانی و هم یک گذر فوری توانا هستند، گذر فوری دارای اولویت بالاتر خواهد بود:



□ اگر در یک نشانه‌گذاری بیش از یک گذر فوری توانا باشد، آنگاه تداخل با استفاده از احتمالات شلیک کردن متناسب شده به گذرهای فوری حل خواهد شد:



## تعریف صوری GSPNs

■ **تعریف ۲:** شبکه پتری تصادفی تعمیم یافته یک ده تایی  $GSPN = (P, T, I, H, O, \Pi, W, PAR, PRED, MP)$  است به نحوی که:

- $P, T, I, H, O, \Pi, PAR, PRED, MP$  مطابق تعریف (۱) هستند.
- $W: T \rightarrow R$ : یک تابع تعریف شده بر روی مجموعه گذرها ( $T$ ) است، به نحوی که  $R$  مجموعه اعداد حقیقی مثبت است.
- تابع  $W$  اجازه تعریف مولفه‌های تصادفی یک مدل GSPN را می‌دهد.
- به بیان دقیق‌تر، تابع  $W$  گذرها را به توابعی از نشانه‌گذاری‌های GSPN که اعداد حقیقی مثبت برمی‌گردانند، نگاشت می‌کند.
- بنا بر این، برای هر گذر  $t \in T$  لازم است که یک تابع  $W(t, M)$  (وابسته به حالت) مشخص شود.
- تابع  $W$  ممکن است که مستقل از حالت نیز باشد، که در این صورت از  $w_k$  برای نشان دادن  $W(t_k)$  استفاده می‌شود که  $t_k \in T$ .
- مقدار  $W(t, M)$  (یا  $w_k$  در حالت مستقل از حالت):
  - اگر  $t_k$  یک گذر زمانی باشد، نرخ (rate) گذر  $t_k$  را در نشانه‌گذاری  $M$  مشخص می‌کند.
  - اگر  $t_k$  یک گذر فوری باشد، وزن (weight) گذر  $t_k$  را در نشانه‌گذاری  $M$  مشخص می‌کند.

## دسته بندی نشانه گذاری های مدل های GSPNs

- در مدل های GSPN به نشانه گذاری هایی که در آنها گذرهای فوری توانا باشند، به دلیل آنکه شلیک کردن این گذرها در زمانی به مدت صفر تمام می شود، **نشانه گذاری های ناپدیدشونده (vanishing markings)** می گویند.
- در مقابل، به نشانه گذاری هایی که در آنها تنها گذرهای زمانی توانا باشند، **نشانه گذاری های محسوس (tangible marking)** می گویند.
- به دلیل آنکه سیستم مدل سازی شده، زمانی به مدت صفر را در نشانه گذاری های ناپدیدشونده صرف می کند، این نشانه گذاری ها در **رفتار زمانی (temporal behavior)** سیستم تاثیر نگذاشته و باید حذف شوند.
- تنها در این صورت است که با روشی مشابه آنچه در مورد SPNs داریم، می توانیم به CTMC برسیم و آنرا حل کنیم.
- این نکته، همان اختلاف عقیده ای بود که مابین ملوی و اجمان مرسان وجود داشت.
- در ادامه خواهیم دید که چگونه اجمان مرسان نشان داد که این کار شدنی است.

## نحوه انتخاب یک گذر برای شلیک کردن در یک نشانه گذاری محسوس

- در یک نشانه گذاری محسوس، انتخاب یکی از گذرهای زمانی جهت شلیک کردن با توجه به **زمان اقامت (sojourn time)** در آن نشانه گذاری انجام می شود:
- از آنجایی که در هر نشانه گذاری محسوس:
  - (۱) همه زمانهای شلیک کردن گذرهای زمانی دارای تابع چگالی احتمالی (pdf) نمایی هستند، و
  - (۲) همه زمانهای شلیک کردن متغیرهای تصادفی مستقل هستند،
- **آنگاه زمان اقامت در یک نشانه گذاری محسوس یک متغیر تصادفی با pdf نمایی منفی (negative exponential) [۱] است که نرخ آن حاصل جمع نرخهای همه گذرهای زمانی توانا در آن نشانه گذاری خواهد بود.**
- این نتیجه از این حقیقت ناشی می شود که حداقل یک مجموعه متغیرهای تصادفی مستقل دارای pdf نمایی منفی دارای یک pdf نمایی منفی دارای نرخ خواهد بود که حاصل جمع نرخ pdf های مجزا است.
- [۱] اگر  $\exp(\lambda)$  یک تابع توزیع نمایی باشد که  $\lambda > 0$  باشد، آنگاه  $\exp(-\lambda)$  یک تابع توزیع نمایی منفی است.

### نحوه انتخاب یک گذر برای شلیک کردن در یک نشانه‌گذاری ناپدید شونده

- اگر گذرهای فوری توانا در یک نشانه‌گذاری ناپدیدشونده دارای سطوح اولویتی متفاوتی باشند، آن گذر فوری که سطح اولویتی بالاتری دارد شلیک خواهد نمود.
- اما اگر با لحاظ کردن سطح اولویت گذرها، بیش از یک گذر فوری متعلق به یک گروه اولویتی در آن نشانه‌گذاری دارای تداخل باشند (یعنی همزمان توانا بوده و بخواهند شلیک کنند)، وزنه‌های گذرهای فوری توانا (همان احتمالات شلیک کردن) برای تعیین یک گذر فوری که شلیک خواهد نمود قابل استفاده است.

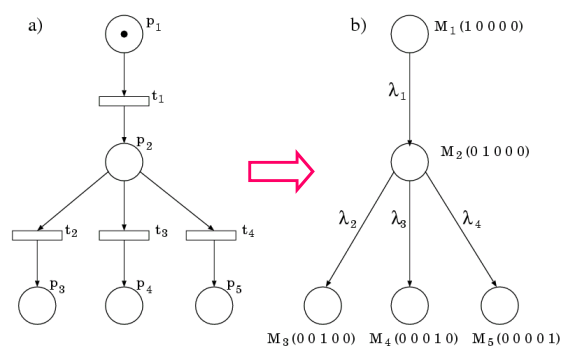
### گراف دسترس‌پذیری مدل‌های GSPNs

- با روشی که قبلاً برای تولید درخت دسترس‌پذیری و تبدیل آن به گراف دسترس‌پذیری معرفی کردیم، می‌توانیم گراف دسترس‌پذیری GSPNs را تولید کنیم.
- به گراف دسترس‌پذیری حاصله، به دلیل آن هم دارای نشانه‌گذاری‌های محسوس و هم ناپدید شونده است، گراف دسترس‌پذیری بسط‌یافته (ERG: extended reachability graph) گفته می‌شود.
- نکته بسیار مهم آن است که این ERG دیگر هم‌ریخت با CTMC نخواهد بود. چرا؟
  - چون گره‌هایی در آن وجود دارند که نرخ گذر از آنها به حالت بعدی صفر است.
- چاره کار چیست و چگونه می‌توانیم به CTMC برسیم؟
  - لازم است که نشانه‌گذاری‌های ناپدید شونده از گراف دسترس‌پذیری بسط‌یافته (ERG) حذف شده و یک گراف دسترس‌پذیری معمولی (RG) حاصل شود، که هم‌ریخت با یک CTMC است. این همان کاری است که اجمان مرسان انجام داد.

## تبدیل ERG به CTMC

- برای تبدیل ERG به CTMC با در نظر گرفتن نشانه‌گذاریهای مختلف (اعم از ناپدیدشونده و محسوس)، سه وضعیت ممکن است پیش بیاید:
  - وضعیت ۱: فقط گذرهای زمانی در یک نشانه‌گذاری توانا باشند.
  - وضعیت ۲: یک گذر فوری و تعدادی گذر زمانی در یک نشانه‌گذاری توانا باشند.
  - وضعیت ۳: بیش از یک گذر فوری در یک نشانه‌گذاری توانا باشند.
- در ادامه با ارائه مثالهایی می‌بینیم که در هر کدام از وضعیت‌های فوق چه کاری باید انجام شود...

## وضعیت ۱: فقط گذرهای زمانی توانا باشند

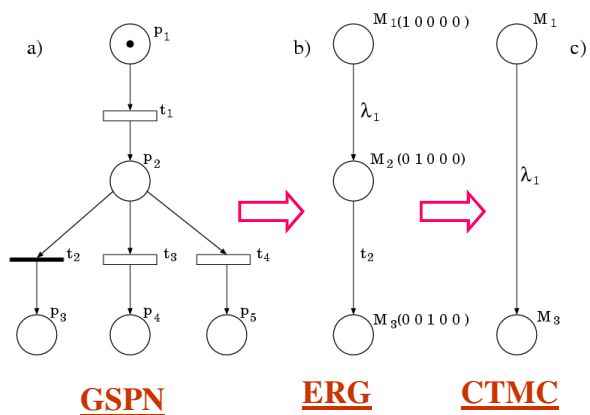


**GSPN**

**RG = CTMC**

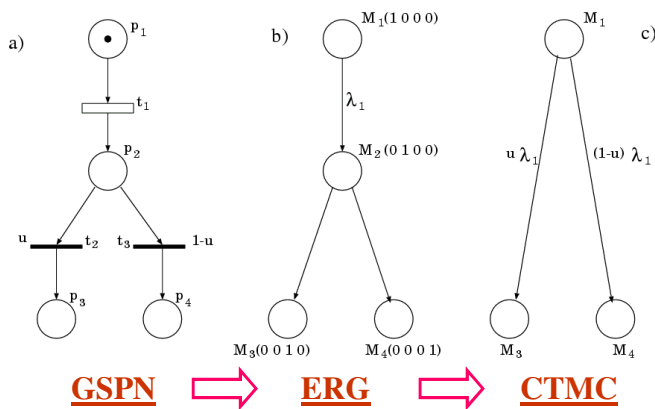
• در این وضعیت، نشانه‌گذاری محسوس بوده و کار خاصی لازم نیست انجام شود.

## وضعیت ۲: یک گذر فوری و تعدادی گذر زمانی توانا باشند



در این وضعیت نشانه گذاری های ناپدید شونده وجود دارد که حذف می شوند.

## وضعیت ۳: بیش از یک گذر فوری توانا باشد



در این وضعیت هم نشانه گذاری های ناپدید شونده وجود دارد که حذف می شوند.



### حل مدل‌های GSPNs

- با تبدیل ERG به RG و بدست آوردن CTMC، حل مدل‌های GSPNs به حل CTMC حاصله تبدیل می‌شود.
- حالا می‌توانیم همانند SPNs، طبق روشهای حل حالت پایدار یا حل گذرا که قبلاً برای CTMC معرفی نمودیم، عمل کنیم.

### محاسبه معیارها از مدل‌های GSPNs

- نکته مهمی که باید بدان توجه شود آن است که مدل‌ساز، معیارهای مورد نظر خود را بر اساس مفاهیم اولیه و اجزاء مدل GSPN تعریف می‌کند و می‌خواهد براساس آنها نیز معیارهای محاسبه شده را بدست آورد، نه بر اساس CTMC:
- مثلاً مدل‌ساز ممکن است که بخواهد میانگین تعداد نشانه‌ها در یک مکان یا توان عملیاتی (یعنی تعداد دفعات شلیک کردن) گذر را بدست آورد.
- بنا بر این اگر نتیجه حل مدل، احتمال حالت پایدار مربوط به یک حالت CTMC باشد، برای مدل‌ساز قابل فهم نخواهد بود.

## محاسبه معیارها از مدل‌های GSPNs

- بنا بر این، یک ابزار مدل‌سازی و ارزیابی باید به مدل‌ساز امکان دهد که معیارهای کارایی مورد نظرش را در سطح شبکه پتری تعریف نموده و نتایج حاصله هم در سطح شبکه پتری محاسبه شود.
- برای مثال می‌توان به معیارهای کارایی زیر اشاره نمود:
  - ☐ احتمال وقوع یک شرایط داده شده در شبکه پتری،
  - ☐ میانگین زمان صرف شده در یک نشانه‌گذاری،
  - ☐ توزیع نشانه‌ها در یک مکان، یا
  - ☐ میانگین تعداد شلیک‌کردن‌های یک گذر (توان عملیاتی).
- ابزارهای مدل‌سازی متعددی نظیر GreatSPN و SHARPE وجود دارند که GSPNs و SPNs را پشتیبانی نموده و امکانات ساخت و حل مدل‌ها و محاسبه معیارها را فراهم می‌کنند.

## شبکه‌های فعالیت تصادفی



- شبکه‌های فعالیت تصادفی (SANs: stochastic activity networks) همزمان با GSPNs در سال ۱۹۸۴ توسط پروفسور علی موقر و تحت راهنمایی پروفسور مایر (J. F. Meyer) در دانشگاه میشیگان برای ارزیابی انجام‌پذیری معرفی شد [1, 2].
- مدل‌های SANs مبنای چندین ابزار مدل‌سازی قوی قرار گرفته و توسط افراد و موسسات متعددی استفاده شده است.
- این مدل‌ها در کاربردهای مختلف برای ارزیابی کارایی، اتکاءپذیری و انجام‌پذیری سیستم‌های متنوعی استفاده شده است.
- سال ۲۰۰۱ تعریف جدیدی از SANs با هدف رفع برخی مشکلات تعریف اولیه توسط دکتر موقر ارائه گردید [3].

- [1] Movaghgar, A. and Meyer, J.F.: "Performability Modeling with Stochastic Activity Networks," *Proc. of the 1984 Real-Time Systems Symp.*, Austin, TX, USA (1984) 215-224
- [2] Movaghgar, A., Performability Modeling with Stochastic Activity Networks, Ph.D. Dissertation, The University of Michigan (1985)
- [3] Movaghgar, A.: "Stochastic Activity Networks: A New Definition and Some Properties," *Scientia Iranica* 8(4) (2001) 303-311

## شبکه‌های فعالیت تصادفی



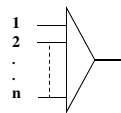
- از جمله افراد معروفی که کارهای فراوانی را مبتنی بر مدل‌های SANs انجام داده، پروفسور سندرز (W. H. Sanders)، استاد دانشگاه ایلی‌نوی است.
- ایشان در دانشگاه‌های آریزونا و ایلی‌نوی کارهای فراوانی را بر اساس مدل‌های SANs انجام داده و دو ابزار معروف UltraSAN و Möbius در **گروه تحقیقاتی مهندسی انجام‌پذیری** [1] را برای مدل‌سازی و ارزیابی کارایی، اتکاءپذیری و انجام‌پذیری طراحی و ساخته است.
- در ادامه به اختصار شبکه‌های فعالیت تصادفی را معرفی می‌کنیم.

[1] "Performability Engineering Research Group," URL: <http://www.perform.csl.uiuc.edu/>

## مفاهیم اولیه تعریف جدید شبکه‌های فعالیت تصادفی

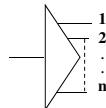
- **مکان (place):** معادل مکانهای شبکه‌های پتری است.
- **فعالیت زمانی (timed activity):** مشابه گذر زمانی GSPNs است:
  - امکان مدل‌سازی **توازی (parallelism)** را فراهم می‌کنند.
  - به هر فعالیت زمانی یک تابع توزیع (عمومی) زمان فعالیت (activity time distribution function)، یک نرخ تواناسازی (enabling rate) (سرعت اجرای فعالیت) و یک گزاره فعالسازی مجدد (reactivation predicate) متناسب می‌شود.
- **فعالیت آنی (instantaneous activity):** مشابه گذر فوری GSPNs است:
  - امکان مدل‌سازی **عدم قطعیت (non-determinism)** را فراهم می‌کنند.
  - به هر فعالیت آنی یک تابع احتمال مورد (case probability function)، متناسب می‌شود که امکان مدل‌سازی احتمالی عدم قطعیت را فراهم می‌کند.

## مفاهیم اولیه تعریف جدید شبکه‌های فعالیت تصادفی



■ **دروازه ورودی (input gate):** تعدادی مکان ورودی را به یک فعالیت متصل می‌کند:

□ برای هر دروازه ورودی امکان تعریف یک گزاره فعال‌سازی و یک تابع ورودی وجود دارد که نشانه‌ها را از مکانهای ورودی برمی‌دارد.



■ **دروازه خروجی (output gate):** یک فعالیت را به تعدادی مکان خروجی متصل می‌کند:

□ برای هر دروازه خروجی امکان تعریف یک تابع خروجی وجود دارد که نشانه‌هایی را به مکانهای خروجی اضافه می‌کند

## خصوصیت مهم شبکه‌های فعالیت تصادفی

■ تعریف جدید شبکه‌های فعالیت تصادفی امکان مدل‌سازی **توازی** (parallelism) و **عدم قطعیت** (non-determinacy) را در سه تنظیم متفاوت فراهم می‌کند:

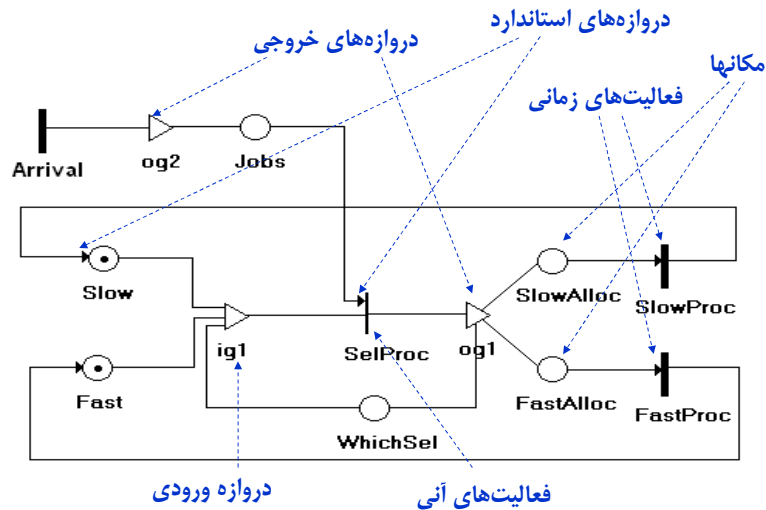
نام تنظیم	عدم قطعیت	توازی	تنظیم
شبکه‌های فعالیت (activity networks)	غیر قطعی	غیر قطعی	غیر قطعی (non-deterministic)
شبکه‌های فعالیت احتمالی (probabilistic activity networks)	احتمالی	غیر قطعی	احتمالی (probabilistic)
شبکه‌های فعالیت تصادفی (SANs: stochastic activity networks)	احتمالی	احتمالی	تصادفی (stochastic)

■ کاربرد تنظیم غیر قطعی در تحلیل عملکردی و درستی‌یابی است،

■ کاربرد تنظیم احتمالی در درستی‌یابی احتمالی (probabilistic verification) است، و

■ کاربرد تنظیم تصادفی در تحلیل عملیاتی و ارزیابی سیستم‌ها است.

## مثالی از شبکه‌های فعالیت تصادفی



PECS 18 - Analysis of Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۷۳

## جدول دروازه مدل

■ برای هر مدل SAN یک جدول دروازه تعریف می‌شود:

Gate Table		
Gate	Predicate	Function
<i>ig1</i>	$(\text{MARK}(\text{Fast}) \neq 0) \parallel (\text{MARK}(\text{Slow}) \neq 0)$	<pre> if (MARK(Fast) != 0) {     MARK(Fast) = 0;     MARK(WhichSel) = 0; } else {     MARK(Slow) = 0;     MARK(WhichSel) = 1; }                     </pre>
<i>og1</i>		<pre> if (MARK(WhichSel) == 0)     MARK(FastAlloc) = 1; else     MARK(SlowAlloc) = 1;                     </pre>
<i>og2</i>		<pre> if (MARK(Jobs) &lt; 10)     MARK(Jobs) = MARK(Jobs)+1;                     </pre>

تعریف شده در یک زبان سطح بالا مثل C/C++

PECS 18 - Analysis of Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۷۴

### قواعد ساخت شبکه‌های فعالیت

- برای ساخت مدل‌های شبکه‌های فعالیت قواعد زیر باید رعایت شود:
  - هر ورودی دروازه ورودی و هر خروجی دروازه خروجی تنها به یک مکان متصل شود،
  - خروجی هر دروازه ورودی یا ورودی هر دروازه خروجی تنها به یک فعالیت متصل شود،
  - دروازه‌های ورودی متصل به یک فعالیت به مکانهای متفاوتی متصل شوند،
  - دروازه‌های خروجی متصل به یک فعالیت به مکانهای متفاوتی متصل شوند،
  - هر فعالیت و هر مکان به دروازه‌های ورودی و خروجی متصل شوند.
- برای سادگی نمایش اگر دروازه ورودی دارای فقط یک ورودی بوده و گزاره تواناسازی آن به صورت  $e(x): x \geq 1$  و تابع خروجی آن به صورت  $f(x) = x - 1$  باشد به شکل یک پیکان ( $\rightarrow$ ) نمایش داده می‌شود.
- در مورد دروازه‌های خروجی که تابع خروجی آن  $f(x) = x + 1$  نیز همین روش استفاده می‌شود. به چنین دروازه‌هایی دروازه های استاندارد گفته می‌شود.

### قواعد اجرای شبکه‌های فعالیت

- شبکه‌های فعالیت سیستم‌های پویایی هستند که در طول زمان اجرا می‌شوند.
- یک اجرای شبکه فعالیت (بدون در نظر گرفتن توزیع تصادفی زمانهای فعالیت‌های زمانی) ترتیبی از نشانه‌گذاری‌ها است که هر کدام با شلیک شدن یک فعالیت به دست می‌آید.
- با شلیک کردن یک فعالیت نشانه‌گذاری‌های شبکه تغییر می‌کند. به دنبال آن مجموعه فعالیت‌های توانا تغییر می‌یابد.
- فعالیت‌های آنی بلافاصله پس از توانا شدن شلیک می‌کنند.
- اما در یک نشانه‌گذاری ناپایدار (unstable marking) که در آن یک یا چند فعالیت آنی توانا هستند، انتخاب یکی از آنها به صورت احتمالی انجام می‌شود.

### قواعد اجرای شبکه‌های فعالیت

- فعالیت‌های زمانی تنها در صورتی قادر به شلیک کردن هستند که در یک نشانه‌گذاری پایدار توانا شده باشند.
- از میان همه فعالیت‌های زمانی که در یک نشانه‌گذاری پایدار توانا شده باشند در یک تنظیم غیرقطعی هر فعالیت به صورت غیرقطعی شلیک می‌کند.
- شلیک کردن یک فعالیت شامل دو مرحله است:
  - اجرای توابع ورودی دروازه‌های ورودی متصل به آن، و
  - اجرای توابع خروجی دروازه‌های خروجی متصل به آن.
- حاصل اجرا، تغییر نشانه‌گذاری شبکه خواهد بود و براساس آن فعالیت‌های دیگری توانا می‌شوند.

### قواعد اجرای شبکه‌های فعالیت تصادفی

- اجرای شبکه‌های فعالیت تصادفی به طور کلی مشابه اجرای شبکه‌های فعالیت است اما دارای تفاوت‌های زیر است:
- هر فعالیت زمانی به سه دلیل می‌تواند فعال شود (activate):
  - بر اثر شلیک شدن فعالیت‌های دیگر در یک نشانه‌گذاری جدید توانا شود
  - در نشانه‌گذاری جاری توانا بماند یا
  - گزاره فعال‌سازی مجدد (reactivation predicate) آن ارزش درست بگیرد.
- پس از فعال شدن یک فعالیت زمانی، یک زمان تصادفی بر اساس توزیع مربوطه که زمان کامل شدن بالقوه (potential completion time) نامیده می‌شود، طول می‌کشد تا شلیک کند.
- اگر قبل از آن زمان ناتوان شود، سقط‌شده (aborted) نامیده شده و نمی‌تواند شلیک کند.

### ویژگیهای شبکه‌های فعالیت تصادفی

- به خاطر وجود امکان تعریف توابع برای دروازه‌های ورودی و خروجی، SANs امکانات مدل‌سازی قوی‌تر و قابل انعطاف‌تری در مقایسه با GSPNs برای مدل کردن سیستم‌ها فراهم نموده است.
- توزیع احتمالی زمانهای فعالیت‌های زمانی در مدل‌های SANs طبق تعریف عمومی است. این در حالی است که در GSPNs فقط توزیع نمایی در نظر گرفته شده است.
- وجود دروازه‌های ورودی و خروجی و امکان تعریف تابع برای آنها و نیز وجود مفاهیم فعال‌سازی مجدد و نرخ تواناسازی باعث تسهیل مدل‌سازی سیستم‌های پیچیده با مدل‌های SANs (در مقایسه با GSPNs) می‌شود.
- به دلیل آنکه توزیع احتمالی زمانهای فعالیت‌های زمانی در مدل‌های SANs عمومی است، در حالت کلی مدل‌های SANs را نمی‌توان با به‌دست آوردن زنجیره مارکوف ادغام‌شده تحلیل نمود. برای مدل‌هایی که دارای توزیع‌های غیرنمایی برای فعالیت‌های زمانی هستند شبیه‌سازی قابل استفاده است.

PECS 18 - Analysis of Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۷۹

### شبکه‌های پتری سطح بالا

- شبکه‌های پتری سطح بالا (high-level Petri nets)، با افزودن **زمان**، **رنگ** و **سلسله‌مراتب** به شبکه‌های پتری بدست می‌آیند:
  - $HLPNs = PNs + Colour + Hierarchy$
- **رنگی بودن:** یعنی نشانه‌ها از هم متمایز هستند. نشانه‌های رنگی، قابلیت پردازش داده‌ها را به مدل اضافه می‌کنند.
- مقایسه شبکه‌های پتری اولیه با شبکه‌های پتری سطح بالا، همانند مقایسه زبان اسمبلی با زبانهای برنامه‌سازی سطح بالا است:
  - در تئوری، هر دو نوع زبان دارای **قدرت محاسباتی** (computational power) یکسانی هستند.
  - اما در عمل، زبانهای سطح بالا دارای **قدرت مدل‌سازی** (modeling power) بالاتری هستند. زیرا قابلیت‌های ساختاردهی و توصیف بهتری را با استفاده از انواع داده‌ای، پیمانه‌ها و غیره فراهم می‌کنند.

PECS 18 - High-Level Petri Nets - By: M. Abdollahi Azgomi - IUST-CE

۸۰



## برخی از شبکه‌های پتری سطح بالا

### ■ شبکه‌های گزاره-گذر (Pr/T-nets: predicate/transition nets)

□ سال ۱۹۷۸ توسط Genrich & Lautenbach معرفی شده است.

### ■ شبکه‌های پتری رنگی (CPNs: coloured Petri nets)

□ سال ۱۹۸۰ توسط K. Jensen معرفی شده است.

### ■ شبکه‌های خوش‌شکل (WNs: well-formed nets):

□ WNs = simple CPNs

### ■ استاندارد HLPN:

□ سال ۲۰۰۳ توسط ISO/IEC معرفی شده است.

## شبکه‌های پتری رنگی



■ شبکه‌های پتری رنگی در سال ۱۹۸۰ توسط کورت ینسن (K. Jensen) معرفی شده است.

■ ینسن استاد دانشگاه آرهوس (Aarhus) دانمارک بوده و سرپرست گروه شبکه‌های پتری رنگی [1] است.

■ در این گروه ابزارهای معروفی نظیر Design/CPN و CPN Tools برای مدل‌سازی با این مدل‌ها طراحی و ساخته شده است.

■ در ادامه مدل‌های CPNs را به اختصار معرفی می‌کنیم.

[1] "CPN Group at the University of Aarhus," <http://www.daimi.au.dk/CPnets/>

### شبکه‌های پتری رنگی

- یک بسط سطح بالا برای شبکه‌های پتری بوده و ترکیبی از شبکه‌های پتری و قابلیت‌های زبانهای برنامه‌سازی سطح بالا است.
- در CPNs نشانه‌ها دارای خصوصیت‌هایی هستند که به آن **رنگ** نشانه گفته می‌شود.
- رنگ نشانه‌ها در حقیقت نشان‌دهنده تعلق داشتن نشانه‌ها به گروه‌های متفاوت و تمایز رفتارشان در شبکه است.
- مفهوم رنگ برای نشانه‌ها مشابه مفهوم زنجیره‌ها (chains) یا رده‌ها (classes) در شبکه‌های صف است.
- هر مکان، یک مجموعه چندگانه (multiset) (یا کیسه) از نشانه‌ها را نگهداری می‌کند.
- هر نشانه یک شیء داده‌ای از یک نوع اتمی (atomic) (مثل integer، char و غیره) یا یک ساختار داده‌ای تعریف شده توسط کاربر است.

### شبکه‌های پتری رنگی

- گذرها با توجه به رنگ نشانه‌های در دسترس در ورودی به صورت متفاوت رخ می‌دهند.
- از عبارتهای **کمان** (arc expressions) و **نگهبان‌ها** (guards) برای تعیین شروط تواناسازی و اثرات رخداد گذرها استفاده می‌شود.
- از زبان ML استاندارد (که یک زبان برنامه‌سازی تابعی است) برای تعریف داده‌ها و عبارتهای شرطی و غیره استفاده می‌شود.
- به دلیل تسهیل مدل‌سازی سیستم‌های واقعی این بسط شبکه‌های پتری بسیار مورد توجه قرار گرفته است.

### شبکه‌های پتری شیء‌گرا

■ فقدان قابلیت‌های ساختاردهی (structuring capabilities) یکی از مهمترین انتقادهای از شبکه‌های پتری سطح بالا بوده است. یعنی زیرمدل‌های آنها فاقد ساختار مناسبی برای ترکیب (composition) هستند.

■ از اینرو محققان اقدام به تقویت شبکه‌های پتری سطح بالا با مفاهیم شیء‌گرایی نمودند و شبکه‌های پتری شیء‌گرا (object-oriented Petri nets) معرفی شدند:

$$\square \text{ OOPNs} = \text{CPNs} + \text{OO}$$



■ از جمله این محققین، چارلز لاکاس (C. Lakos) است که استاد دانشگاه آدلاید استرالیا است.

■ در ادامه به اختصار شبکه‌های پتری شیء‌گرا معرفی می‌شود.

### شبکه‌های پتری شیء‌گرا

■ در OOPNs مفاهیم شیء‌گرایی با مفاهیم شبکه‌های پتری ترکیب یا جایگزین شده است.

■ برای مثال مفهوم مکان با ساختار فیلد داده‌ای (data fields) یا خصوصیت‌های (attributes) شیء و گذرها با مفهوم متد یا تابع جایگزین شده‌اند.

■ نخستین بسط مهم OOPNs نسبت به CPNs این است که امکان تعریف مدل به صورت رده (class) در آن وجود دارد:

□ این رده‌ها قابل نمونه‌سازی (instantiation) خواهند بود.

□ هر رده مدل می‌تواند تعدادی فیلد داده‌ای و تابع داشته باشد.

□ OOPNs توارث (inheritance) را برای رده‌ها پشتیبانی می‌کند.

■ یک فیلد داده‌ای می‌تواند یک نوع ساده (مثل Integer، Real، Boolean، و غیره)، نوع رده یا از نوع یک مجموعه چندگانه از این انواع باشد.

### شبکه‌های پتری شیء‌گرا

- امکان تعریف توابع با قابلیت دسترسی فقط خواندنی (read-only) به حالت جاری زیرشبکه یکی دیگر از تفاوت‌های OPNs با CPNs است.
- این مساله باعث افزایش قابلیت بسته‌بندی (encapsulation) در OPNs می‌شود.
- مدل‌های OPNs از نظر رفتاری معادل مدل‌های CPNs هستند. یعنی برای تحلیل می‌توان OPNs را به CPNs تبدیل نمود.
- این مدل‌ها برای طراحی شیء‌گرای سیستم‌های نرم‌افزاری مناسب هستند و شاید بتوان از برخی جهات و در بعضی کاربردها آنها را به عنوان یک آلترناتیو UML مورد استفاده قرار داد.
- از مزایای OPNs آن است که قابلیت ترکیب (compositionality) مدل‌ها و مدل‌سازی تدریجی (incremental) و سلسله‌مراتبی (hierarchical) را فراهم می‌کنند.

### سایر بسط‌های شبکه‌های پتری

- بسط‌های متعدد دیگری هم برای شبکه‌های پتری معرفی شده است که در ادامه چهار مورد از آنها به اختصار معرفی می‌شوند:
- شبکه‌های پاداش تصادفی (SRNs: stochastic reward nets)
- شبکه‌های پتری تصادفی تجدیدپذیر مارکوف (MRSPNs: Markov regenerative SPNs)
- شبکه‌های پتری تصادفی سیال (FSPNs: fluid stochastic Petri nets)
- شبکه‌های فعالیت تصادفی سطح بالا (high-level SANs)

### شبکه‌های پاداش تصادفی (SRNs)

- یکی از بسط‌های GSPNs است که در آن به نشانه‌گذاری‌ها، پاداش‌های نرخ (rate rewards) مرتبط می‌شود.
- این پاداش‌های نرخ در مدت زمان اقامت SPNs در یک نشانه‌گذاری جمع‌آوری می‌شود.
- کاربرد این مدل در مدل‌سازی اتکاء‌پذیری و سیستم‌های تحمل‌پذیر خطا است.

### شبکه‌های پتری تصادفی تجدیدپذیر مارکوف (MRSPNs)

- مدل‌های تجدیدپذیر حالت کلی‌تر مدل‌های مارکوف هستند.
- در این مدل‌ها زمان‌های بین ورود از یک توزیع مستقل یکسان (iid: independent identical distribution) پیروی می‌کنند.
- MRSPNs رده‌ای از SPNs است که در آن زمان‌های منتسب شده به گذرها می‌تواند یک توزیع مستقل یکسان باشد.

### شبکه‌های پتری تصادفی سیال (FSPNs)

- یکی دیگر از بسط‌های GSPNs است که در آن برخی از مکانها می‌توانند به جای مقادیر گسسته و از نوع عدد صحیح، دارای مقادیر پیوسته مثبت باشند، شبکه‌های پتری تصادفی سیال هستند.
- بخشهای پیوسته و گسسته مدل می‌توانند با هم ارتباط داشته باشند.
- به این ترتیب فرآیند مارکوف حاصله دارای حالت پیوسته خواهد بود.
- هدف این مدل مقابله با مشکل انفجار فضای حالت (state space explosion) است.

### شبکه‌های فعالیت تصادفی سطح بالا

- اخیراً سه بسط سطح بالا برای مدل‌های SANs معرفی شده‌اند که امکانات مدل‌سازی سلسله‌مراتبی، پردازش داده‌ها و شیء‌گرایی را فراهم می‌کنند. این بسطها عبارتند از [1]:

- شبکه‌های فعالیت تصادفی سلسله‌مراتبی (HSANs: hierarchical SANs)
  - HSANs = Hierarchy + SANs
- شبکه‌های فعالیت تصادفی رنگی (CSANs: coloured SANs)
  - CSANs = Hierarchy + Colour + SANs
- شبکه‌های فعالیت تصادفی شیئی (OSANs: object SANs)
  - OSANs = Hierarchy + Colour + OO + SANs

[1] Abdollahi Azgomi, M., *High-Level Extensions for Stochastic Activity Networks: Theories, Tools and Applications*, Ph.D. Dissertation, Department of Computer Engineering, Sharif University of Technology (2005)

## نمودار گسترش و تکامل بسط‌های شبکه‌های پتری

