

Lecture 18: Petri-Nets (Continued)

Kenneth M. Anderson
Foundations of Software Engineering
CSCI 5828 - Spring Semester, 1999

Today's Lecture

- Finish the Filling Station Example
- Look at analysis techniques using Petri Nets
- Look at extensions to the basic Petri Net formalism
 - add “data” to tokens
 - add “conditionals” to transitions

Filling Station Example

- Lets model the following situation
 - Fuel Pumps
 - Spaces next to Pumps
 - A cashier that takes payment
- Questions
 - What is the concurrency that we want modeled?
 - How do we handle the parameterization of the Petri net? (e.g. lets say I want to add a pump)

Concurrency Problems

- Starvation
 - Enabled transition never fired
- Deadlock
 - Unintended lack of enabled transitions
- V&V Tries to Detect These Problems
 - Static and dynamic analysis techniques

Shortcoming of Basic Petri Nets

Simplicity of building blocks leads to complexity in nets

Example: Semaphore for n processes requires $2n$ transitions and $3n+1$ places

Would Like...

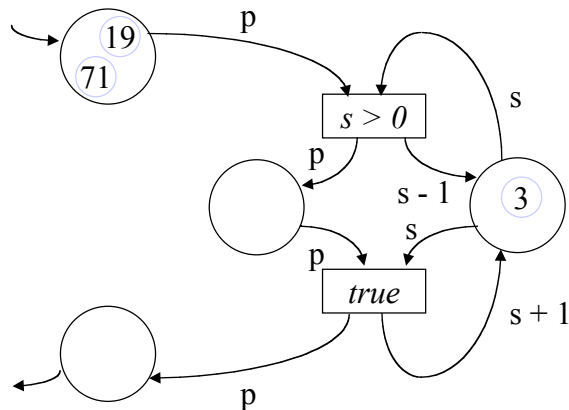
- Enable and fire as computations
- Tokens as data, not just control

Higher-Level Petri Nets

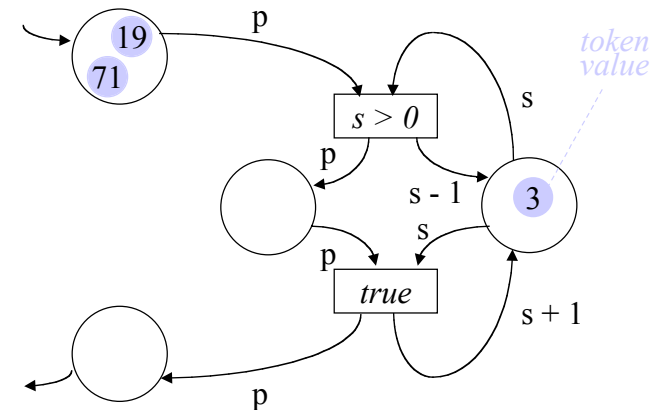
- Some Enhancements to Basic Petri Nets
 - Typed places and information-bearing tokens
 - Predicate transitions
 - Hierarchical decomposition of places and transitions

Requirement for analysis of higher-level nets: reducible to basic nets for analysis

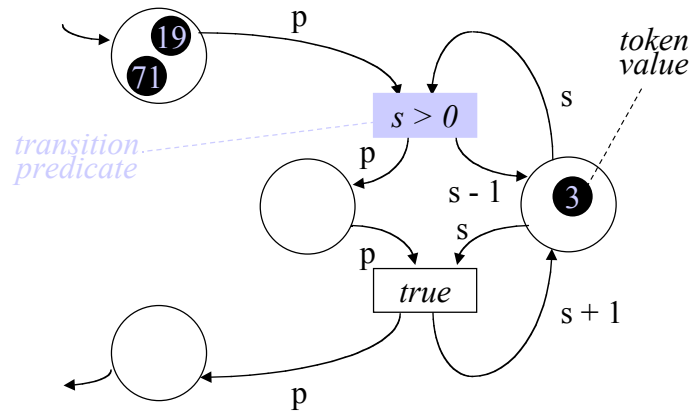
Higher-Level Net



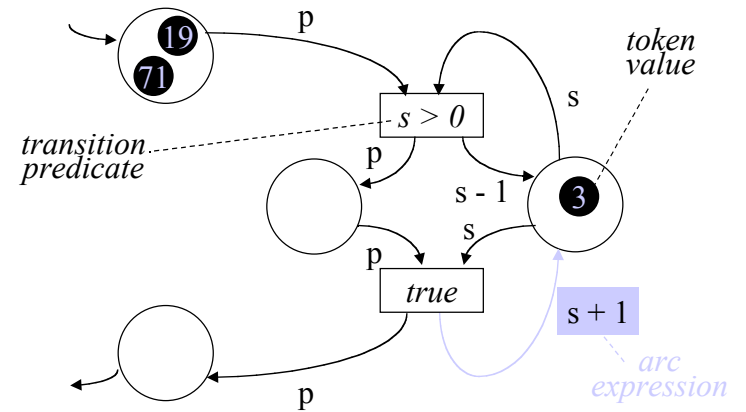
Higher-Level Net



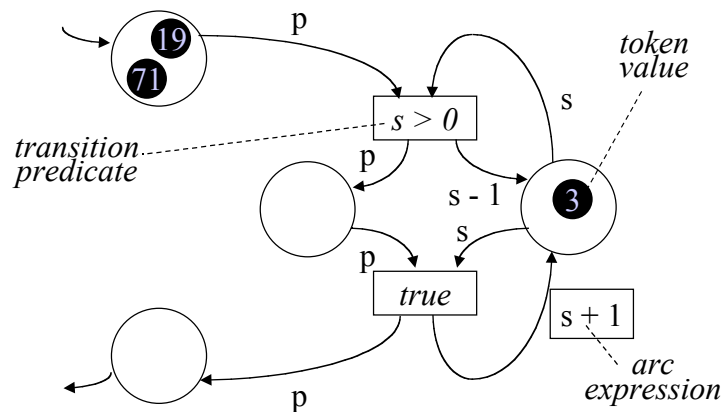
Higher-Level Net



Higher-Level Net



Higher-Level Net



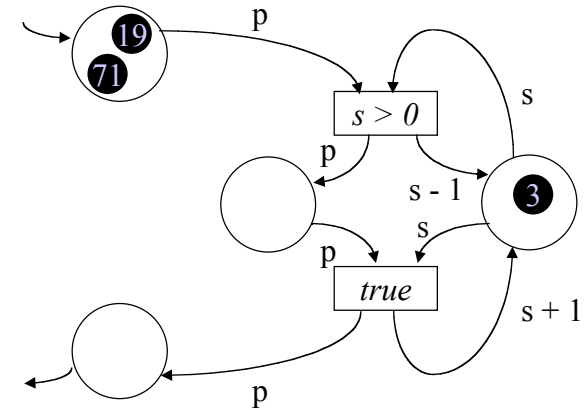
Execution Model

- “Enable” is a Predicate on Input Tokens
 - Transition with k input places is enabled if there exists a k -tuple of tokens, one at each input place, that satisfy the predicate; called a *ready tuple*
 - Enabled transition and ready tuple are nondeterministically selected
 - Tokens of selected ready tuple removed at firing

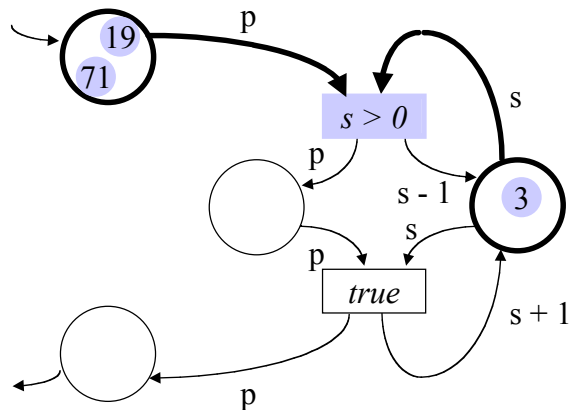
Execution Model

- Function Computes Output Token Values
 - Transition with h output places uses the function to compute h values, one for each output token

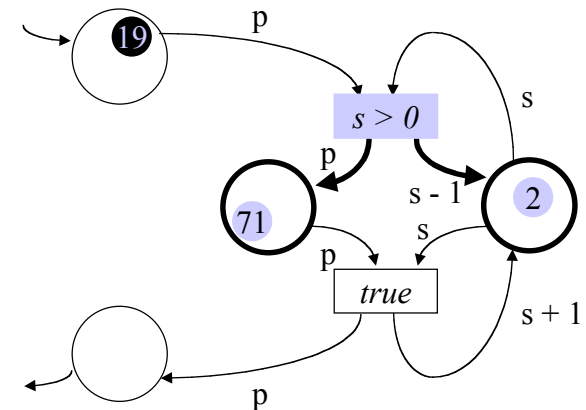
Higher-Level Net Semaphore



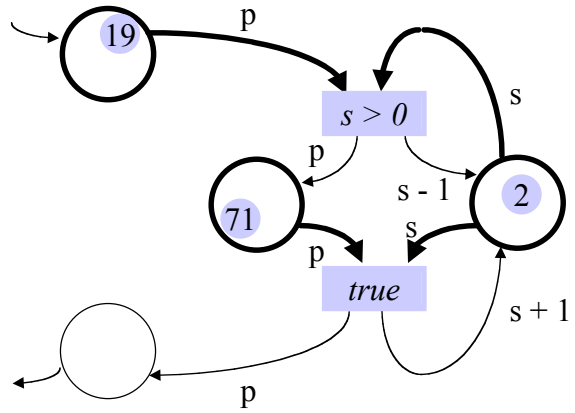
Enabled Transition



After Firing



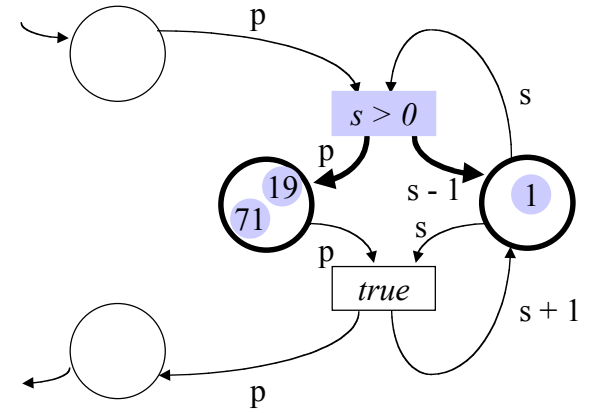
Enabled Transitions



Lecture 18

17

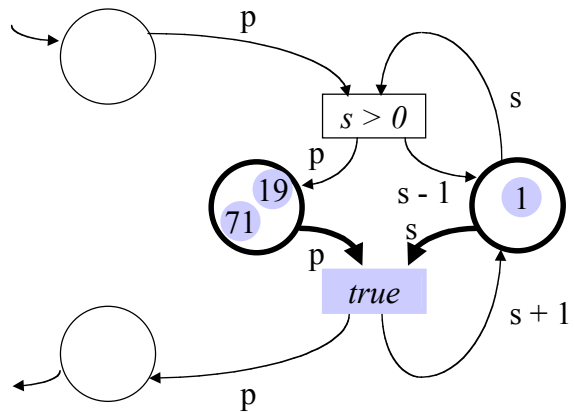
After Firing



Lecture 18

18

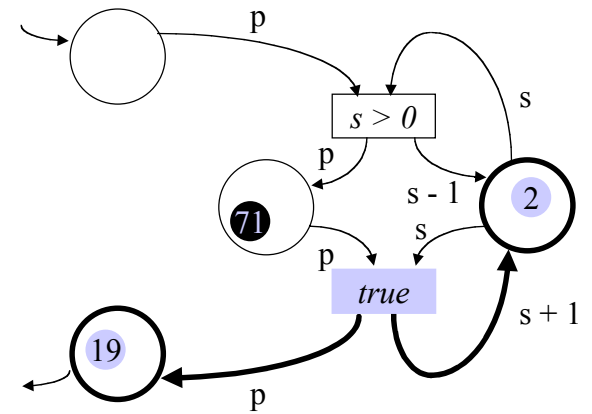
Enabled Transition



Lecture 18

19

After Firing



Lecture 18

20