ILP Formulation of Scheduling

Dr. Chandan Karfa

Department of Computer Science and Engineering

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

Preprocessing

Example: 2^{nd} order differential equation solver

```
Diffeq: (x, dx, u, a, clock, y)
input: x, dx, u, a, clock;
output: y
while(x < a)
        u1 = u-(3*x*u*dx)-(3*y*dx);
        y1 = y+(u*dx);
        x1 = x+dx;
    x = x1, y = y1, u = u1;
end
```

I

Read(p1, dx)

Read(p2, x)

Read(p3, a)

Read(p1,y)

Read(p2, u)

c = x < a

B2

Write(p1, y)

B_1

$V_1 : t_1 = u * dx$

$V_2 : t_2 = 3 * x$

$V_3 : t_3 = 3 * y$

$V_4 : t_4 = u * dx$

$V_5 : t_5 = t_1 * t_2$

$V_6 : t_6 = t_3 * dx$

$V_7 : t_7 = u – t_5$

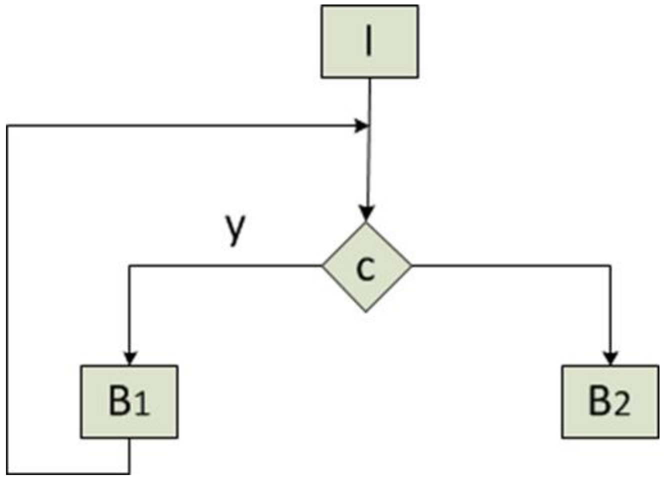
$V_8 : u = t_7 – t_6$

$V_9 : y = y + t_4$

$V_{10} : x = x + dx$

$V_{11} : c = x < a$

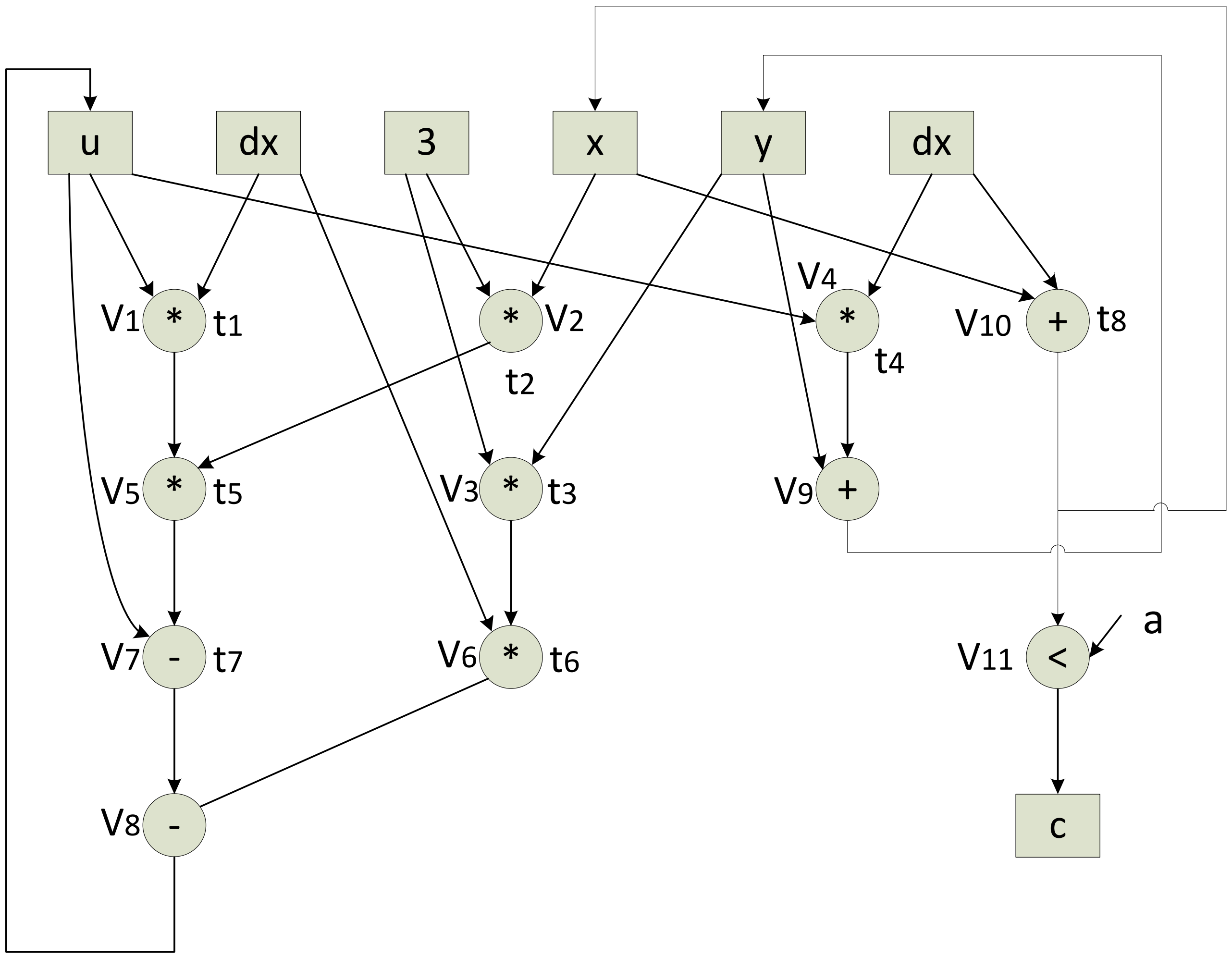
Basic Blocks with 3-address codes



Control and Dataflow graph (CDFG)

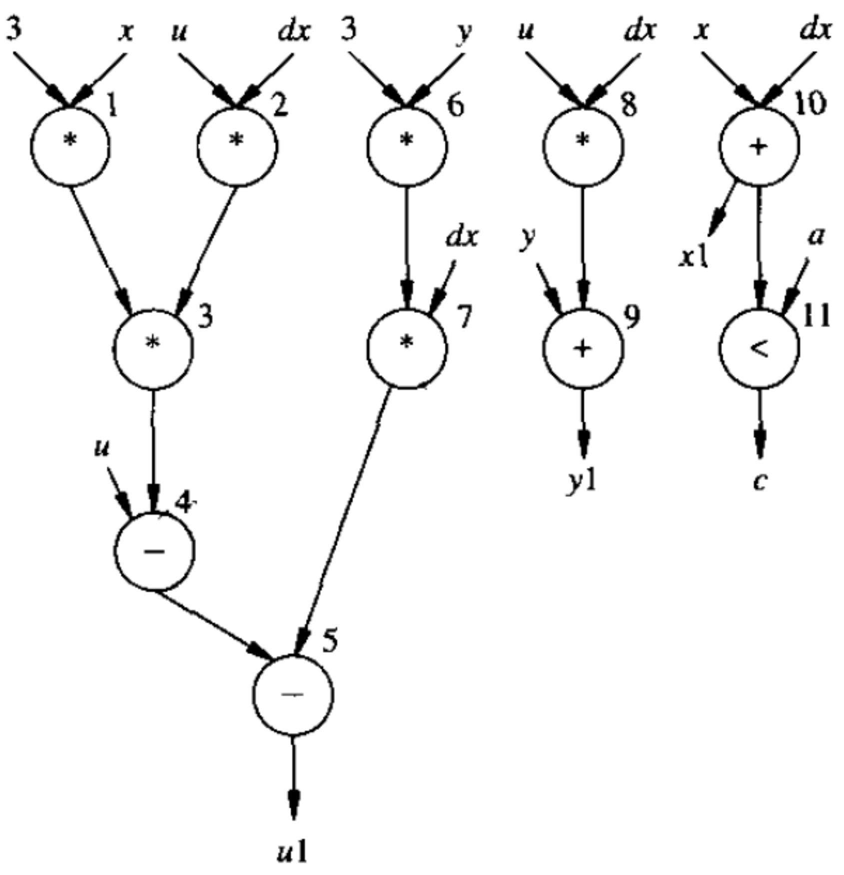
Preprocessing

B_1
$V_1 : t_1 = u * dx$
$V_2 : t_2 = 3 * x$
$V_3 : t_3 = 3 * y$
$V_4 : t_4 = u * dx$
$V_5 : t_5 = t_1 * t_2$
$V_6 : t_6 = t_3 * dx$
$V_7 : t_7 = u - t_5$
$V_8 : u = t_7 - t_6$
$V_9 : y = y + t_4$
$V_{10} : x = x + dx$
$V_{11} : c = x < a$

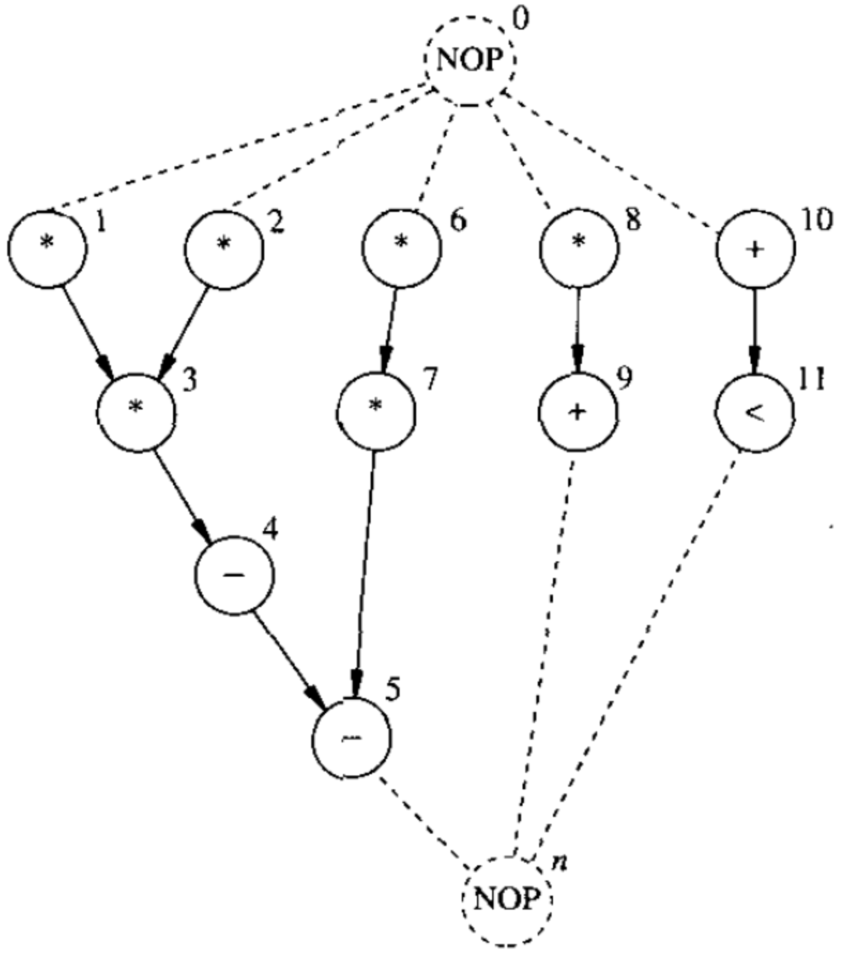


Date dependency graph

Sequence Graph

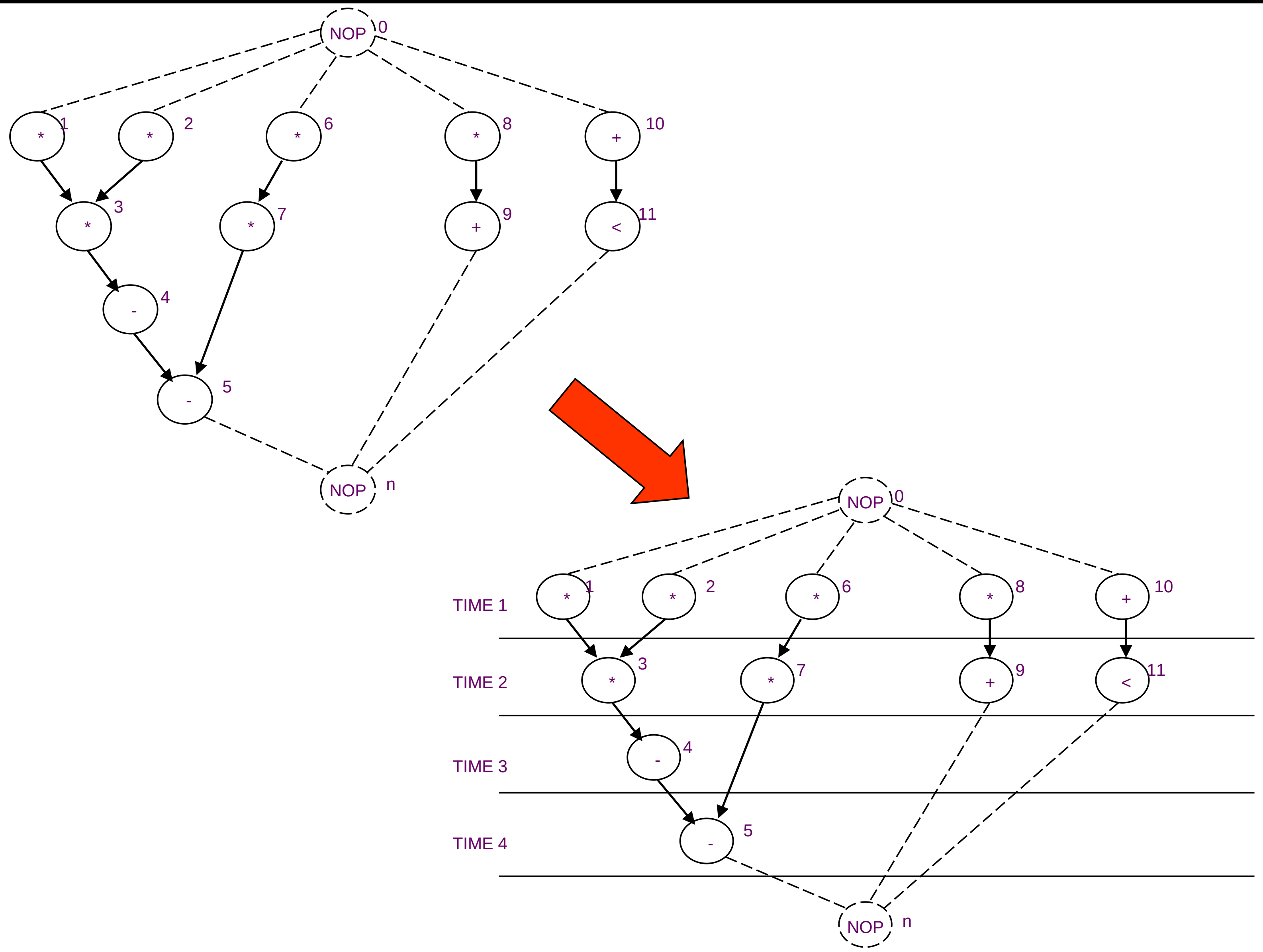


Dataflow graph



Sequence Graph

Scheduling



Scheduling Problem Formulation

Input:

- Sequence Graph G = (V, E), |V| = n
- Delay of each node. D = {d_i, i = 0, 1, …, n}
- Resource or Timing Constraints (optional)

Output:

- The start time of each node T = {t_i, i=0, 1, 2, …, n}
- Latency: number of cycles to execute the entire schedule. Difference of start time of source node and sink node; latency = $t_n - t_0$

The start time of an operation is at least as large as the start time of each of its direct predecessor plus its execution delay

$$t_i \geq t_j + d_j \quad \forall\, i, j \quad : \; (v_j, v_i) \in E$$

Scheduling Problems

- Minimum Latency Unconstrained minimum-latency scheduling problem (Unconstraint)
- Minimum latency under resource constraints (MLRC)
- Minimum resource under latency constraints (MRLC)

Scheduling under resource/timing constraints

- NP Complete Problem
- Algorithms:
  - Exact:
    - Integer linear program
    - Hu (restrictive assumptions)
  - Approximate/Heuristic based:
    - List scheduling
    - Force-directed scheduling

Integer Linear Programming (ILP)

- Given:
  - integer-valued matrix $\boxed{A_{m \times n}}$
  - *variables:* $\mathbf{x} = (x_1, x_2, \ldots , x_n)^T$
  - constants: $\mathbf{b} = (b_1, b_2, \ldots , b_m)^T$ and $\mathbf{c} = (c_1, c_2, \ldots , c_n)^T$

m → restrictions
n → no of nodes

- Minimize: $\boxed{c^T \mathbf{x}}$

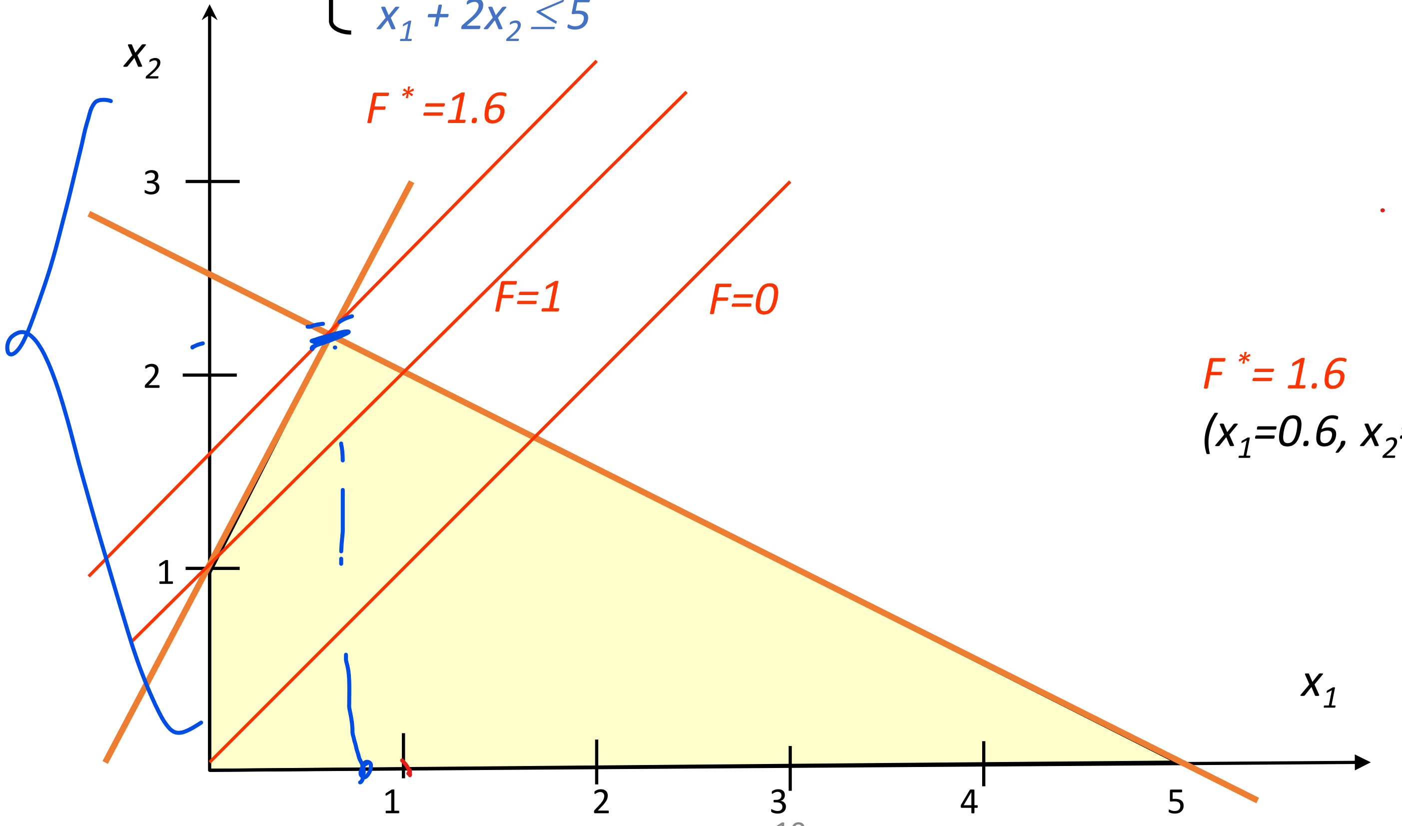
subject to:

$$\begin{cases} \boxed{A \mathbf{x} \le \mathbf{b}} \\ \mathbf{x} = (x_1, x_2, \ldots , x_n) \text{ is an integer-valued vector} \end{cases}$$

- If all variables are *continuous*, the problem is called linear (LP)
- Problem is called *Integer* LP (ILP) if some variables x are integer
  - special case: 0,1 (binary) ILP

Linear Programming – example

- Variables: $x = [x_1, x_2]^T$
- Objective function: max $F = -x_1 + x_2 = [-1\ 1]\ [x_1, x_2]^T$
- Constraints: $\begin{cases} -2x_1 + x_2 \le 1 \\ x_1 + 2x_2 \le 5 \end{cases}$



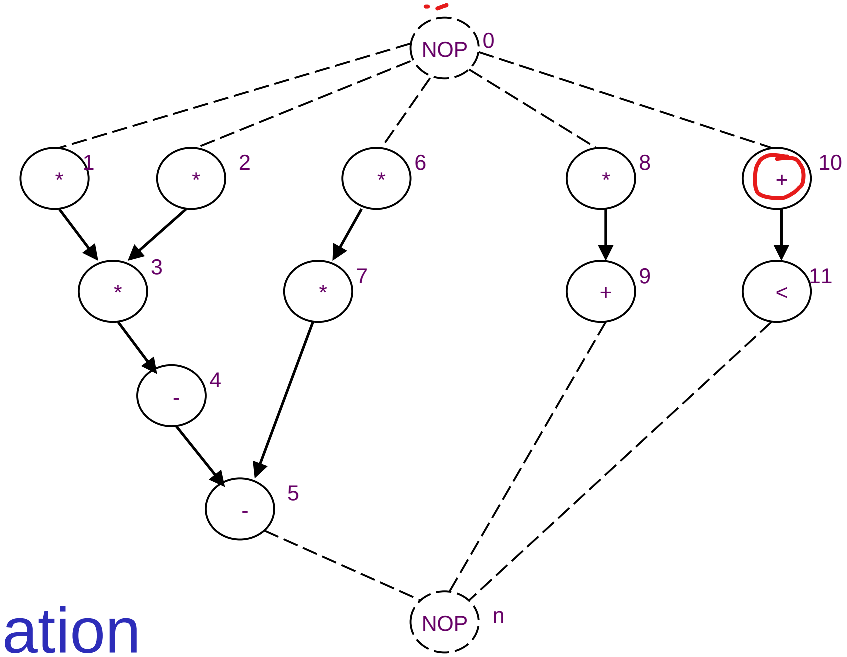
$F^* = 1.6$

$(x_1 = 0.6,\ x_2 = 2.2)$

ILP Model of Scheduling

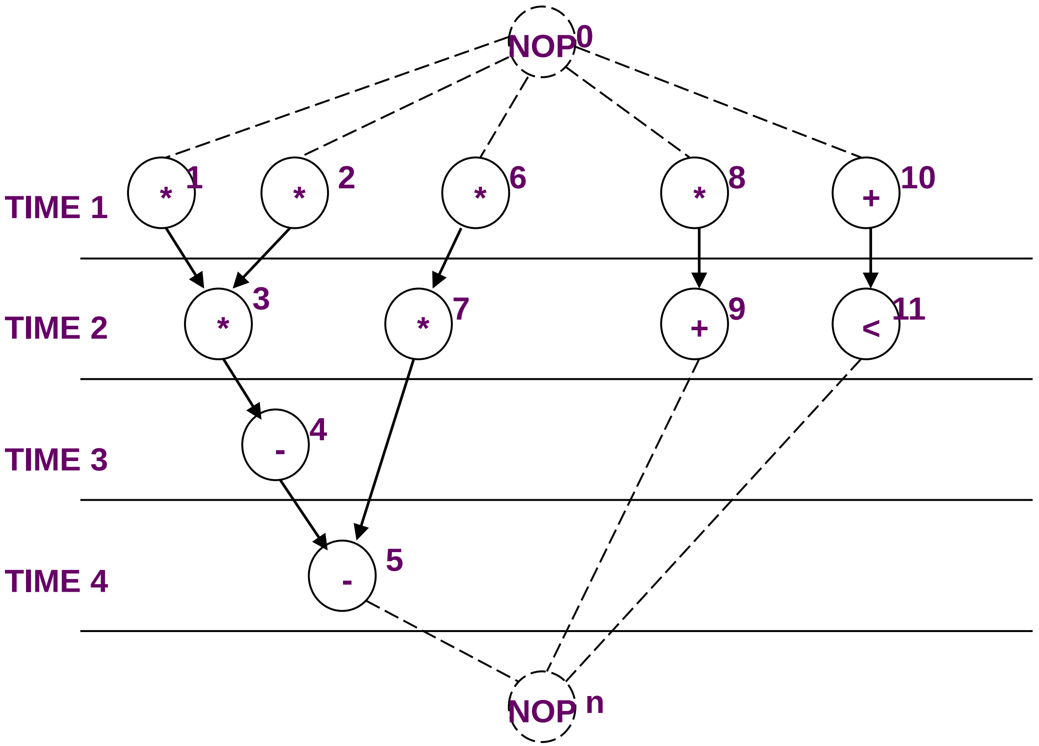
- Binary decision variables x_{il}

  $x_{il} = 1$ if operation v_i starts in step l,
  otherwise $x_{il} = 0$
  $i = 0, 1, \ldots, n$ (operations)
  $l = 1, 2, \ldots \lambda+1$ (steps, with limit λ)

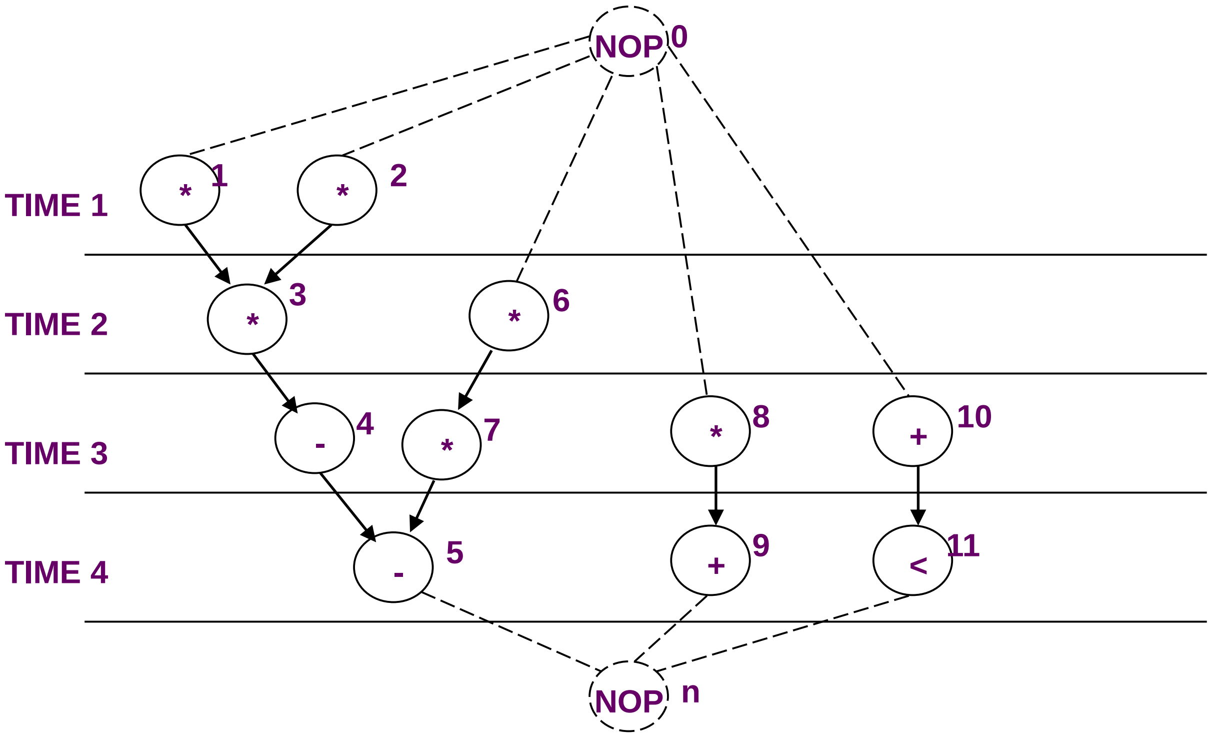


Q1: How many Binary variable do we need?
Q2: Can we use ASAP and ALAP scheduling information reduce the decision variable?

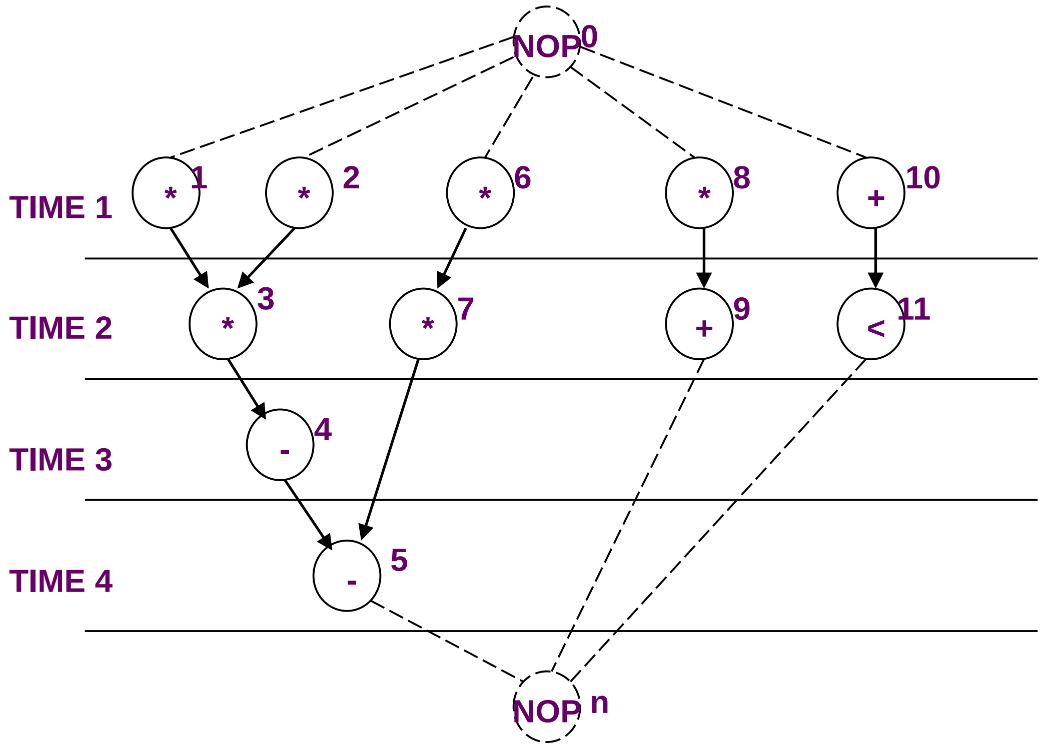
Mobility



ASAP



ALAP


NOP 0
* 1
* 2
* 6
* 8
+ 10
TIME 1
* 3
* 7
+ 9
< 11
TIME 2
- 4
TIME 3
- 5
TIME 4
NOP n

ILP Model of Scheduling - Constraints

- Start time of each operation v_i is unique:

$$\sum_l x_{il} \;=\; 1, \quad i = 0, 1, \dots, n$$

$V_{10} \rightarrow \{1, 2, 3\}$

$V_{11} \{2, 3, 4\}$

Note: $\sum_l x_{il} = \sum_{l=t_i^S}^{l=t_i^L} x_{il}$

where:

t_i^S = time of operation I computed with *ASAP*

t_i^L = time of operation I computed with *ALAP*

Start time for v_i:

$$t_i = \sum_l l \cdot x_{il}$$

x_{il}: $x_{10\,1} = 0$, $x_{10\,2} = 1$, $x_{10\,3} = 0$

$t_{10} = 1 \times 0 + 2 \times 1 + 3 \times 0$

2

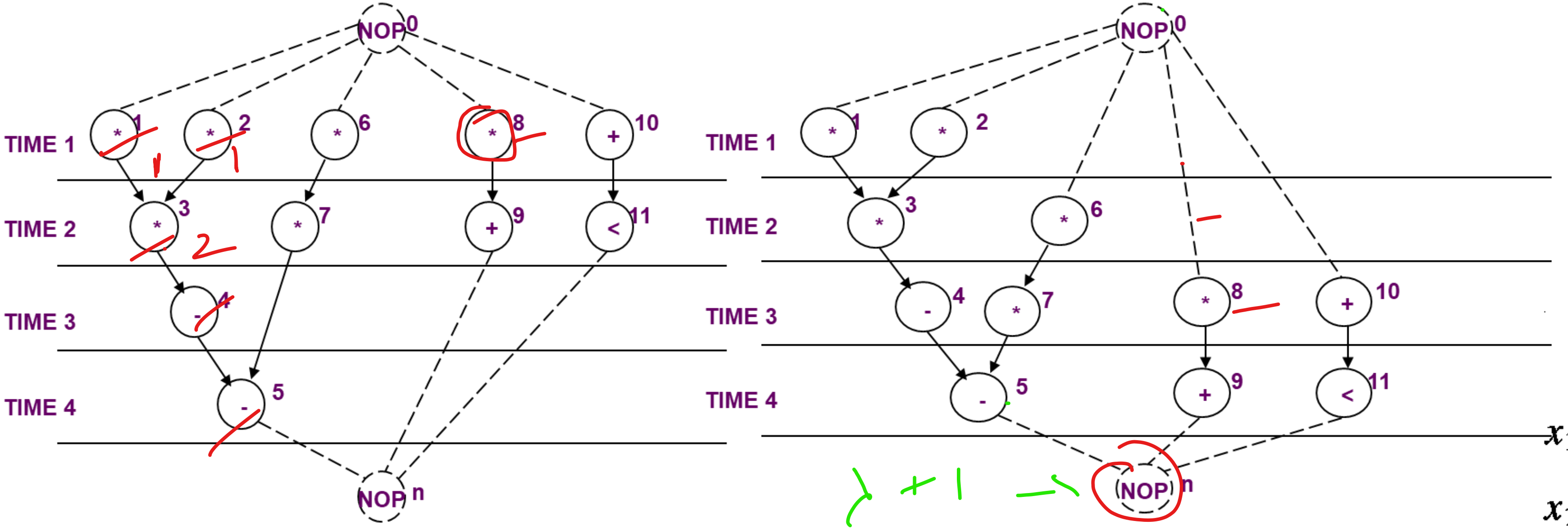
$t_{10} = \sum_{l=t_i^S}^{t_i^L} l \times x_{il}$

Example 1 (cont'd.)

- Start time must be unique

Recall: $\sum_l x_{il} = \sum_{l=t_i^S}^{t_i^L} x_{il}$

x_8



$$
\begin{aligned}
x_{0,1} &= 1 \\
x_{1,1} &= 1 \\
x_{2,1} &= 1 \\
x_{3,2} &= 1 \\
x_{4,3} &= 1 \\
x_{5,4} &= 1 \\
x_{6,1} + x_{6,2} &= 1 \\
x_{7,2} + x_{7,3} &= 1 \\
x_{8,1} + x_{8,2} + x_{8,3} &= 1 \\
x_{9,2} + x_{9,3} + x_{9,4} &= 1 \\
x_{10,1} + x_{10,2} + x_{10,3} &= 1 \\
x_{11,2} + x_{11,3} + x_{11,4} &= 1 \\
x_{n,5} &= 1
\end{aligned}
$$

λ+1

ILP Model of Scheduling - constraints

- Precedence relationships must be satisfied

$$\sum_l l \cdot x_{il} \geq \sum_l l \cdot x_{jl} + d_j, \quad i, j = 0, 1, \ldots, n \quad : (v_j, v_i) \in E$$

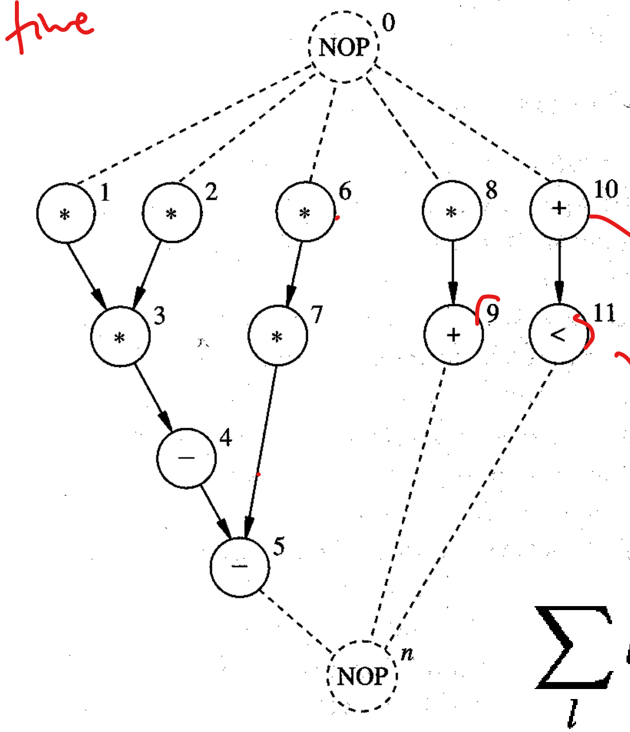
Example 1 (cont'd.)

- Precedence constraints
  - Note: only non-trivial ones listed

x_{il} → time

↓

vertex



$$2x_{7,2} + 3x_{7,3} - x_{6,1} - 2x_{6,2} - 1 \geq 0$$

$$2x_{9,2} + 3x_{9,3} + 4x_{9,4} - x_{8,1} - 2x_{8,2} - 3x_{8,3} - 1 \geq 0$$

$$2x_{11,2} + 3x_{11,3} + 4x_{11,4} - x_{10,1} - 2x_{10,2} - 3x_{10,3} - 1 \geq 0$$

$$4x_{5,4} - 2x_{7,2} - 3x_{7,3} - 1 \geq 0$$

$$5x_{n,5} - 2x_{9,2} - 3x_{9,3} - 4x_{9,4} - 1 \geq 0$$

$$5x_{n,5} - 2x_{11,2} - 3x_{11,3} - 4x_{11,4} - 1 \geq 0$$

$$\sum_l \underset{t_i}{l \cdot x_{il}} \geq \underbrace{\sum_l l \cdot x_{jl}}_{t_j} + d_j, \quad i, j = 0, 1, \ldots, n \quad : (v_j, v_i) \in E$$

ILP Model of Scheduling - constraints

- Resource constraints must be met
  - let upper bound on number of resources of type k be a_k

$$\sum_{i:\mathcal{T}(v_i)=k} \sum_{m=l-d_i+1}^{l} x_{im} \leq a_k, \quad k = 1, 2, \ldots, n_{res}, \quad l = 1, 2, \ldots, \bar{\lambda} + 1$$

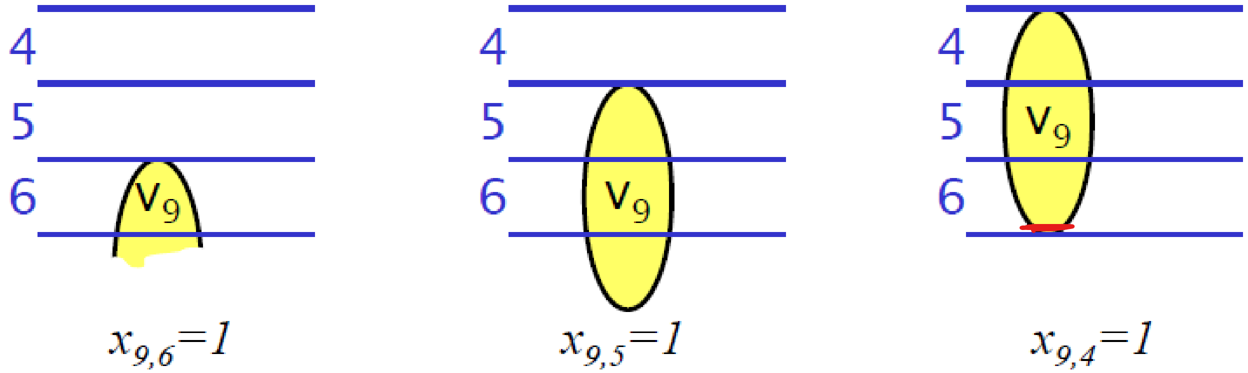
Start Time vs. Execution Time

- For each operation v_i, only one start time
- If $d_i=1$, then the following questions are the same:
  - Does operation v_i **start** at step l?
  - Is operation v_i **running** at step l?
- But if $d_i>1$, then the two questions should be formulated as:
  - Does operation v_i **start** at step l?
    - Does $x_{il} = 1$ hold?
  - Is operation v_i **running** at step l?
    - Does the following hold?

$$\sum_{m=l-d_i+1}^{l} x_{im} \stackrel{?}{=} 1$$

Operation v_i Still Running at Step l ?

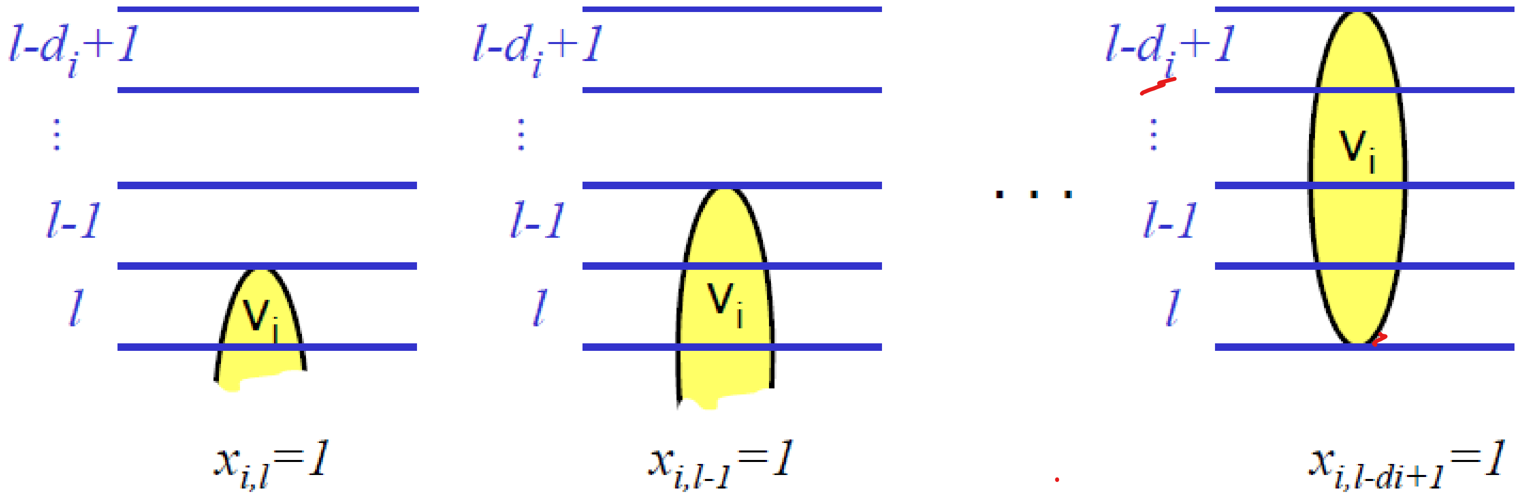
- Is v_9 running at step 6?
  - Is $x_{9,6} + x_{9,5} + x_{9,4} = 1$?



- Note:
  - Only one (if any) of the above three cases can happen
  - To meet resource constraints, we have to ask the same question for ALL steps, and ALL operations of that type

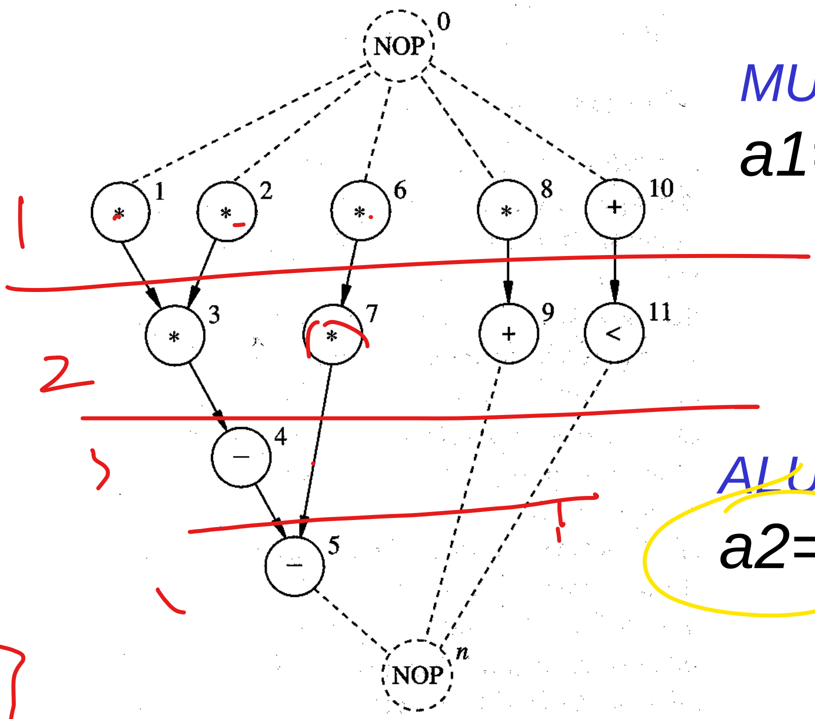
Operation v_i Still Running at Step l ?

- Is v_i running at step l ?
  - Is $x_{i,l} + x_{i,l-1} + \ldots + x_{i,l-di+1} = 1$?



Example 1 (cont'd.)

- Resource constraints



MULT

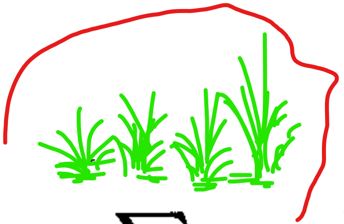
a1=2

$$\begin{cases} x_{1,1} + x_{2,1} + x_{6,1} + x_{8,1} \leq 2 \\ x_{3,2} + x_{6,2} + x_{7,2} + x_{8,2} \leq 2 \\ x_{7,3} + x_{8,3} \leq 2 \end{cases}$$

ALU

a2=2

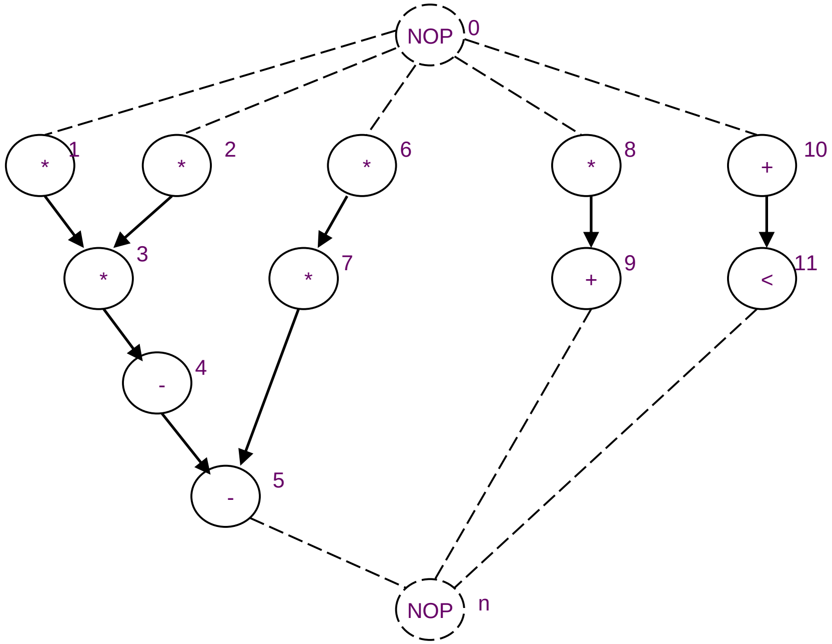
$$\begin{cases} x_{10,1} \leq 2 \\ x_{9,2} + x_{10,2} + x_{11,2} \leq 2 \\ x_{4,3} + x_{9,3} + x_{10,3} + x_{11,3} \leq 2 \\ x_{5,4} + x_{9,4} + x_{11,4} \leq 2 \end{cases}$$



$$\sum_{i:\mathcal{T}(v_i)=k} \; \sum_{m=l-d_i+1}^{l} x_{im} \leq a_k, \quad k = 1, 2, \ldots, n_{res}, \quad l = 1, 2, \ldots, \bar{\lambda} + 1$$

ILP Model of Scheduling - Constraints

- Latency bound must be satisfied
- $\sum_l l \, x_{nl} \leq \lambda + 1$



Scheduling - Objective Function ML RC

- Function to be minimized: $F = c^T t$, *where* $t_i = \sum_l l \cdot x_{il}$
- Minimum latency schedule: $\mathbf{c} = [0, 0, \ldots, 1]^T$
  - $F = t_n = \sum_l l \, x_{nl}$
  - if sink has no mobility ($x_{n,s} = 1$), any feasible schedule is optimum
- ASAP: $\mathbf{c} = [1, 1, \ldots, 1]^T$
  - finds earliest start times for all operations $\sum_i \sum_l x_{il}$
  - or equivalently:

$$x_{6,1} + 2x_{6,2} + 2x_{7,2} + 3x_{7,3} + x_{8,1} + 2x_{8,2} + 3x_{8,3} + 2x_{9,2} + 3x_{9,3} + \\ + 4x_{9,4} + x_{10,1} + 2x_{10,2} + 3x_{10,3} + 2x_{11,2} + 3x_{11,3} + 4x_{11,4}$$

Thank You