Three useful global (complex) constraints will be highlighted.

- All-Different Constraint
- Element Constraint
- Counting Constraint
The All-Different Constraint

- The constrained enumerated variables of array must take different values from each other when they are bound.

```c
IlcConstraint IlcAllDiff(const IlcIntVarArray array, IlcWhenEvent event=IlcWhenValueChanged);
```
int main(int argc, char** argv) {
    IlcManager m(IlcNoEdit);
    IlcInt nqueen = (argc > 1) ? atoi(argv[1]) : 100;
    IlcIntVarArray x(m, nqueen, 0, nqueen-1),
                      x1(m, nqueen), x2(m, nqueen);

    for (int i = 0; i < nqueen; i++) {
        x1[i] = x[i]+i;
        x2[i] = x[i]-i;
    }
    m.add(IlcAllDiff(x));
    m.add(IlcAllDiff(x1));
    m.add(IlcAllDiff(x2));

    m.add(IlcGenerate(x));
    if (m.nextSolution()) m.out() << x << endl;
    else m.out() << "No solution" << endl;

    m.end();
    return 0;
}
When it is necessary to associate a variable with other variable, the element constraint is a likely candidate.

\[ \text{element}(I, [V_1, \ldots, V_n], X) \]

The Element constraint in ILOG can be defined as following,

\[ X = V[I]; \]

where the \( V[] \) can either be an IlcIntVarArray or int[].
The counting constraint to put limits on the number of times that a value can appear in a given array of variables.

Three types:
- `AtMost(n, [X_1, ..., X_n], v)`
- `AtLeast(n, [X_1, ..., X_n], v)`
- `Exactly(n, [X_1, ..., X_n], v)`
The Counting Constraint - 2

- They can be defined by using the class `IlcIndex` and the function `IlcCard`.

- **AtMost:**
  
  \[(\text{IlcCard}(I, X[I] == v) \leq n)\];

- **AtLeast:**
  
  \[(\text{IlcCard}(I, X[I] == v) \geq n)\];

- **Exactly:**
  
  \[(\text{IlcCard}(I, X[I] == v) == n)\];

Where `I` is `IlcIndex`. 
Yet Another Counting Constraint

- **IlcDistribute** is used to count the number of variables that take a given value in an array (unlike **IlcCard**).

```c
IlcConstraint IlcDistribute(
    IlcIntVarArray cards, IlcIntArray values,
    IlcIntVarArray vars,
    IlcFilterLevel level=IlcBasic);

IlcConstraint IlcDistribute(
    IlcIntVarArray cards, IlcAnyArray values,
    IlcAnyVarArray vars,
    IlcFilterLevel level=IlcBasic);

IlcConstraint IlcDistribute(
    IlcIntVarArray cards, IlcIntVarArray vars,
    IlcFilterLevel level=IlcBasic);
```
- The arrays cards and values MUST be the same length.

- The argument level can take either of two values: IlcBasic or IlcExtended.

- IlcExtended causes more domain reduction than IlcBasic, but it takes longer to run.
The following two code fragments are equivalent, but the \texttt{IlcDistribute} is more efficient.

```c
m.add(IlcDistribute(cards, values, vars))
IlcInt i;
IlcIndex j;
IlcInt size = cards.getSize();
for(i=0; i<size; i++)
  m.add(cards[i] == IlcCard(j, vars[j] == values[i]));
```
Nurse Rostering Problem

- To decide which shift of each nurse should work each day in order to satisfy the forecast work demand and employees’ workload.
CSP Modeling...

- **Variables:**
  - Shifts for each nurse in each day, $S_{ij}$

- **Domain:**
  - All possible shifts
  - i.e. { AM, PM, Over-Night, Day-off }

- **Constraints:**
  - Daily constraints
  - Weekly constraints
# A Roster Duty Sheet

## Roster Sheet

<table>
<thead>
<tr>
<th>Name</th>
<th>Sun</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thr</th>
<th>Fri</th>
<th>Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter</td>
<td>$S_{11}$</td>
<td>$S_{12}$</td>
<td>$S_{13}$</td>
<td>$S_{14}$</td>
<td>$S_{15}$</td>
<td>$S_{16}$</td>
<td>$S_{17}$</td>
</tr>
<tr>
<td>Paul</td>
<td>$S_{21}$</td>
<td>$S_{22}$</td>
<td>$S_{23}$</td>
<td>$S_{24}$</td>
<td>$S_{25}$</td>
<td>$S_{26}$</td>
<td>$S_{27}$</td>
</tr>
<tr>
<td>Mary</td>
<td>$S_{31}$</td>
<td>$S_{32}$</td>
<td>$S_{33}$</td>
<td>$S_{34}$</td>
<td>$S_{35}$</td>
<td>$S_{36}$</td>
<td>$S_{37}$</td>
</tr>
<tr>
<td>John</td>
<td>$S_{41}$</td>
<td>$S_{42}$</td>
<td>$S_{43}$</td>
<td>$S_{44}$</td>
<td>$S_{45}$</td>
<td>$S_{46}$</td>
<td>$S_{47}$</td>
</tr>
<tr>
<td>Louis</td>
<td>$S_{51}$</td>
<td>$S_{52}$</td>
<td>$S_{53}$</td>
<td>$S_{54}$</td>
<td>$S_{55}$</td>
<td>$S_{56}$</td>
<td>$S_{57}$</td>
</tr>
</tbody>
</table>

## Weekly Constraint

<table>
<thead>
<tr>
<th>A (AM)</th>
<th>P (PM)</th>
<th>N (Night)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

## Daily Constraint

<table>
<thead>
<tr>
<th>A (AM)</th>
<th>P (PM)</th>
<th>N (Night)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

**CSC5240 Topics in Constraint Processing** 13
### The Daily constraints

On Sunday, there must be at least 3 nurses in the afternoon.

\[ \text{IlcCard}(I, \text{SUN}[I] == 2) >= 3; \]

Where AM:1, PM:2, OverNight:3
### The Daily constraints

#### Daily Constraint

<table>
<thead>
<tr>
<th>Name</th>
<th>Sun</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thr</th>
<th>Fri</th>
<th>Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter</td>
<td>$S_{11}$</td>
<td>$S_{12}$</td>
<td>$S_{13}$</td>
<td>$S_{14}$</td>
<td>$S_{15}$</td>
<td>$S_{16}$</td>
<td>$S_{17}$</td>
</tr>
<tr>
<td>Paul</td>
<td>$S_{21}$</td>
<td>$S_{22}$</td>
<td>$S_{23}$</td>
<td>$S_{24}$</td>
<td>$S_{25}$</td>
<td>$S_{26}$</td>
<td>$S_{27}$</td>
</tr>
<tr>
<td>Mary</td>
<td>$S_{31}$</td>
<td>$S_{32}$</td>
<td>$S_{33}$</td>
<td>$S_{34}$</td>
<td>$S_{35}$</td>
<td>$S_{36}$</td>
<td>$S_{37}$</td>
</tr>
<tr>
<td>John</td>
<td>$S_{41}$</td>
<td>$S_{42}$</td>
<td>$S_{43}$</td>
<td>$S_{44}$</td>
<td>$S_{45}$</td>
<td>$S_{46}$</td>
<td>$S_{47}$</td>
</tr>
<tr>
<td>Louis</td>
<td>$S_{51}$</td>
<td>$S_{52}$</td>
<td>$S_{53}$</td>
<td>$S_{54}$</td>
<td>$S_{55}$</td>
<td>$S_{56}$</td>
<td>$S_{57}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weekly Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

- On Wed, there must be only one nurse during overnight.
- $\text{IlcCard}(I, \text{WED}[I] == 3) == 1$;
- Where AM:1, PM:2, OverNight:3
Paul can work at most four morning shifts per week.

\[
\text{IlcCard}(I, \text{PAUL}[I] == 1) \leq 4;
\]

Where AM:1, PM:2, OverNight:3
Any Questions...