CSC5240: Tutorial #11

Spencer Fung

26 Nov, 2003
Searching for Solutions

- How Solver carries out a search?
  - Which is different from the conventional tree search.

- The implementation of search algorithms is based on the idea of goals in ILOG Solver.

- The search tree in Solver is mainly constructed by two predefined goals: *IlcGenerate* and *IlcInstantiate*. 
IlcGenerate

IlcGoal IlcGenerate(const IlcIntVarArray,
    IlcChooseIntIndex chooseVariable=IlcChooseFirstUnboundInt);

IlcGoal IlcGenerate(const IlcIntVarArray,
    IlcChooseIntIndex chooseVariable, IlcIntSelect sel);

IlcGoal IlcGenerate(const IlcAnyVarArray x,
    IlcChooseAnyIndex chooseVariable=IlcChooseFirstUnboundAny);

IlcGoal IlcGenerate(const IlcAnyVarArray x,
    IlcChooseAnyIndex chooseVariable, IlcAnySelect sel);

IlcGoal IlcGenerate(const IlcIntSetVarArray,
    IlcChooseIntSetIndex chooseVariable=IlcChooseFirstUnboundIntSet);

...etc
What does \texttt{IlcGenerate} do?

- It takes an array of constrained variables as its argument.
- Execution of \texttt{IlcGenerate} has the effect of giving a value to each variable in that.
- More precisely, the goal \texttt{IlcGenerate} exploits the following algorithm:
  1. Choose one of those variables;
  2. Select a value and assign to this variable;
  3. Propagate the effects of that assignment.
Parameters of $\text{IlcGenerate}$

- Parameters for controlling both the choice of variables and the choice of their values.

- The default variable ordering strategy is $\text{IlcChooseFirstUnboundInt}$. 
IlcInstantiate

IlcGoal IlcInstantiate(const IlcIntVar var);
IlcGoal IlcInstantiate(const IlcIntVar var,
                     IlcIntSelect select);
IlcGoal IlcInstantiate(const IlcAnyVar var);
IlcGoal IlcInstantiate(const IlcAnyVar var,
                     IlcAnySelect select);
In the previous tutorial slides...

```c
int main()
{
    IlcManager m(IlcNoEdit);
    IlcIntVar x(m, -3, 5);
    m.add(IlcInstantiate(x, mySelect(m)));
    while (m.nextSolution())
    {
        m.out() << x;
    }
    m.end();
    return 0;
}

/* Output
*/
```
What does \texttt{IlcInstantiate} do?

- The \texttt{IlcInstantiate} function takes a constrained variable as its argument and binds it.
- More precisely, this function \textit{selects a value} in the domain of the constrained variable and \textit{assigns it to the constrained variable}.
- Solver then automatically propagates the constraints posted on this constrained variable.
The default value ordering strategy is to select the minimum value of the variable.

How to construct the search tree by using these two predefined Goals?
Choice point is known as each node in the search tree. They are implemented in Solver by the function

```c
IlcGoal IlcOr(IlcGoal g1, IlcGoal g2);
```

A choice point defines a goal in terms of a choice between sub-goals.

Solver executes a choice point between two sub-goals like this:

1. Save the state of Solver (including the state of all variables and constraints), so that it can be restored if needed.
2. The first sub-goal is executed.
3. If the execution of the first sub-goal leads to an inconsistency, the state of Solver is restored (backtracking), and the second sub-goal is executed.
Implement \texttt{IlcInstantiate} as a Choice Point

1. If the constrained variable is bound (that is, it has already been assigned a value), execution is finished.

2. Otherwise, set a choice point between two sub-goals.
   i. The first sub-goal assigns the minimum domain value of the constrained variable to it.

3. If a contradiction is detected, execute the second sub-goal.
   i. The second sub-goal removes the tried and failed value from the domain of the constrained variable;
   ii. \texttt{calls IlcInstantiate} again.
ILCGOAL1(IlcInstantiate, IlcIntVar, var) {
    if (var.isBound())
        return 0;
    IlcInt value = var.getMin();
    return IlcOr(var == value,
                 IlcAnd(var != value, this));
}

- **ILCGOAL1** is a macro which is used to defines a goal class with single data member.

Note: The actual implementation of **IlcInstantiate** is slightly different since the choice of the value can be indicated by a parameter.
ILCGOAL3(IlcGenerate, IlcIntVarArray, vars, IlcChooseIntIndex, chooseIndex)

IlcInt index = chooseIndex(vars);
if(index == -1)
    return 0;
return IlcAnd(IlcInstantiate(vars[index]), this);

- **ILCGOAL3** is a macro which is used to defines a goal class with 3 data member.

Note: The actual implementation of IlcGenerate is more complex since IlcInstantiate can accept an additional argument, but this version highlights the fundamental ideas of the implementation.
Any Questions...