

Aplicație de Organizare a Orarului folosind Forward Checking

Olaru Ciprian-Timotei, grupa 1410A

1. Descrierea problemei

Proiectul propune realizarea unei aplicații software pentru organizarea unui orar, utilizând algoritmul **Forward Checking**. Scopul aplicației este de a aloca grupe studenților în săli disponibile, respectând o serie de constrângeri specifice, cum ar fi:

- **Disponibilitatea sălilor:** fiecare sală are intervale orare care pot fi ocupate.
- **Disponibilitatea grupelor:** fiecare grupă are propriile intervale orare preferate.
- **Evitarea suprapunerilor:** o sală nu poate fi alocată la mai multe grupe în același interval orar.
- **Condiții de unicitate:** fiecare grupă are un nume unic.

Aplicația permite adăugarea manuală de săli și grupe, generarea automată a orarului pe baza constrângerilor, rezolvarea problemelor prin utilizarea algoritmului de **Forward Checking**, vizualizarea și gestionarea orarului generat și resetarea orarului și ștergerea rezervărilor din săli.

2. Aspecte teoretice

Forward Checking este un algoritm utilizat în rezolvarea problemelor de tip CSP (Constraint Satisfaction Problems).

Caracteristici principale ale algoritmului:

1. **Identificarea soluțiilor valide** - algoritmul verifică posibilitatea alocării unei valori (interval) pentru o variabilă (grupă) fără a încălca constrângerile impuse de alte variabile.
2. **Pruning (tăierea domeniilor invalide)** - după fiecare alocare, elimină din domeniile celorlalte variabile valorile care nu mai sunt posibile.
3. **Backtracking controlat** - algoritmul revine la pasul anterior doar dacă nu mai există soluții valide pentru variabila curentă.
4. **Eficiență** - comparativ cu alte metode (ex. Backtracking), Forward Checking reduce spațiul de căutare, fiind mai rapid în cazul problemelor complexe cu multe constrângeri.

3. Modalitatea de rezolvare

Pentru realizarea aplicației, am folosit algoritmul Forward Checking în următorii pași:

1. **Structurarea datelor:**
 - Sălile sunt reprezentate printr-o clasă Sala, care include numele sălii și intervalele orare rezervate.
 - Grupele sunt reprezentate ca tuple (string grupa, List<(int Start, int End)> intervale).
 - Orarul generat este o listă de tuple (string Grupa, string Sala, (int Start, int End) Interval).
2. **Implementarea funcțiilor principale:**
 - **Adăugarea de săli și grupe:** utilizatorii pot introduce manual datele.
 - **Generarea orarului:** funcția RezolvaOrar alocă grupe în săli folosind Forward Checking.
 - **Validarea:** funcția EsteValid verifică dacă o sală poate fi alocată pentru o anumită grupă într-un interval dat.
3. **Interfața grafică:**
 - Aplicația utilizează Windows Forms pentru afișarea datelor și interacțiunea cu utilizatorul.
 - DataGridView este folosit pentru vizualizarea orarului generat.
4. **Resetare:**
 - Rezervările sunt șterse împreună cu orarul prin funcția btnStergeOrar_Click.

4. Listarea părților semnificative din codul sursă

Adăugarea grupelor:

```
C/C++
private void btnAdaugaGrupa_Click(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(txtGrupa.Text) &&
        !string.IsNullOrEmpty(txtIntervale.Text))
    {
        if (grupe.Any(g => g.grupa.Equals(txtGrupa.Text,
            StringComparison.OrdinalIgnoreCase)))
        {
            MessageBox.Show($"Grupa {txtGrupa.Text} exista deja", "Eroare",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }

        var intervale = new List<(int Start, int End)>();
        foreach (var intervalString in txtIntervale.Text.Split(','))
        {
            var bounds = intervalString.Trim().Split('-');
            if (bounds.Length == 2 && int.TryParse(bounds[0], out int start) &&
                int.TryParse(bounds[1], out int end))
```

```

        {
            intervale.Add((start, end));
        }
    }

    grupe.Add((txtGrupa.Text, intervale));
    listGrupe.Items.Add($" {txtGrupa.Text} (Intervale: {string.Join(" ",
    intervale.Select(i => $"({i.Start}, {i.End})"))});");
    }
}

```

Generarea orarului cu Forward Checking:

```

C/C++
private bool RezolvaOrar(List<(string Grupa, string Sala, (int Start, int End)
Interval)> orar, List<(string grupa, List<(int Start, int End)> intervale)> grupe,
List<Sala> sali)
{
    if (grupe.Count == 0) return true;

    var grupaCurenta = grupe[0];
    var grupaRamase = grupe.GetRange(1, grupe.Count - 1);

    foreach (var interval in grupaCurenta.intervale)
    {
        foreach (var sala in sali)
        {
            if (EsteValid(orar, grupaCurenta.grupa, sala, interval.Start, interval.End))
            {
                orar.Add((grupaCurenta.grupa, sala.Nume, interval));
                RezervaSala(sala, interval.Start, interval.End);

                if (RezolvaOrar(orar, grupaRamase, sali))
                    return true;

                orar.RemoveAll(o => o.Grupa == grupaCurenta.grupa && o.Sala ==
sala.Nume && o.Interval == interval);
                sala.IntervaleRezervate.RemoveAll(i => i.Start == interval.Start && i.End
== interval.End);
            }
        }
    }

    return false;
}

```

Funcție de validare:

C/C++

```
private bool EsteValid(List<(string Grupa, string Sala, (int Start, int End) Interval)>
orar, string grupa, Sala sala, int intervalStart, int intervalEnd)
{
    // verificare interval
    foreach (var rezervare in sala.IntervaleRezervate)
    {
        if (rezervare.Start < intervalEnd && intervalStart < rezervare.End)
            return false;
    }

    // verificare grupa la un alt interval
    foreach (var o in orar)
    {
        if (o.Grupa == grupa && o.Interval.Start < intervalEnd && intervalStart <
o.Interval.End)
            return false;
    }

    return true;
}
```

5. Rezultatele obținute

Interfața grafică a aplicației:

The screenshot shows a Windows application window titled "Form1". On the left, there is a table with three columns: "Grupa", "Sala", and "Interval". The first row has a blue header, and the second row has a grey background. Below the table are two buttons: "Genereaza orar" and "Sterge orar". To the right of the table, there are two large empty rectangular boxes. Below these boxes, there are input fields for "Grupa", "Sala", and "Intervale grupa". There are also buttons for "Adauga grupa", "Sterge Grupe", "Adauga sala", "Rezerva sala", and "Sterge Sali". At the bottom of the window, there are buttons for "Exemple", "Exemplu 1", "Exemplu 2", and "Exemplu 3".

Adăugarea grupelor:

Diagram illustrating the process of adding a group to a reservation.

Left Panel (Initial State):

- Reservation box: Empty
- Grupa:
- Intervale grupa:
- Buttons: (disabled), (disabled)

Right Panel (After Action):

- Reservation box: 1410A (Intervale: (11, 13))
- Grupa:
- Intervale grupa:
- Buttons: (active/highlighted), (disabled)

Adăugarea sălilor:

Diagram illustrating the process of adding a room to a reservation.

Left Panel (Initial State):

- Reservation box: Empty
- Sala:
- Buttons: (disabled), (disabled)
- Rezervare section: De la, Pana la, (disabled)

Right Panel (After Action):

- Reservation box: C0-2 (Libera)
- Sala:
- Buttons: (active/highlighted), (disabled)
- Rezervare section: De la, Pana la, (disabled)

Rezervarea unei săli:

The diagram illustrates the reservation process for room C0-2. It shows two states: the initial state and the reserved state.

Initial State (Left):

- Room selection: C0-2 (Libera)
- Buttons: Sala, Adauga sala
- Reservation period: De la 13, Pana la 15
- Buttons: Rezerva sala, Sterge Sali

Reserved State (Right):

- Room selection: C0-2 (Rezervata: 13-15)
- Buttons: Sala, Adauga sala
- Reservation period: De la 13, Pana la 15
- Buttons: Rezerva sala, Sterge Sali

An arrow points from the initial state to the reserved state, indicating the transition.

Exemplu de generare orar:

The screenshot shows a software interface for generating a schedule. It includes a table of groups, a list of rooms, and various input fields and buttons.

Table:

	Grupa	Sala	Interval
▶	1408A	C0-1	11-13
	1409A	C0-1	13-15
	1410A	C0-1	15-17
*			

Room Selection:

- 1408A (Intervale: (11, 13), (13, 15))
- 1409A (Intervale: (13, 15), (15, 17))
- 1410A (Intervale: (15, 17), (17, 19))
- C0-1 (Libera)

Input Fields and Buttons:

- Grupa:
- Sala:
- Intervale grupa:
- Rezervare: De la Pana la
- Buttons: Adauga sala, Adauga grupa, Sterge Grupe, Rezerva sala, Sterge Sali

Buttons:

- Genereaza orar
- Sterge orar
- Exemple: Exemplu 1, Exemplu 2, Exemplu 3

Exemplu de generare orar cu sală rezervată:

Form1

	Grupa	Sala	Interval
▶	1408A	C0-1	13-15
	1409A	C0-1	15-17
	1410A	C0-1	17-19
*			

1408A (Intervale: (11, 13), (13, 15))
1409A (Intervale: (13, 15), (15, 17))
1410A (Intervale: (15, 17), (17, 19))

C0-1 (Rezervata: 11-13)

Grupa

Sala

Intervale grupa

Rezervare De la Pana la

Adauga grupa

Adauga sala

Sterge Grupe

Sterge Sali

Rezerva sala

Genereaza orar

Sterge orar

Exemple Exemplu 1 Exemplu 2 Exemplu 3

Exemplu de caz în care nu s-a găsit nicio soluție:

Form1

	Grupa	Sala	Interval
*			

1408A (Intervale: (11, 13), (13, 15))
1409A (Intervale: (13, 15), (15, 17))
1410A (Intervale: (15, 17), (17, 19))
1408B (Intervale: (11, 13))

C0-1 (Rezervata: 11-13)

Grupa

Sala

Intervale grupa

Rezervare De la Pana la

Adauga grupa

Adauga sala

Sterge Grupe

Sterge Sali

Rezerva sala

Genereaza orar

Sterge orar

Exemple Exemplu 1 Exemplu 2 Exemplu 3

Eroare

Nu s-a gasit nicio solutie valida.

OK

Exemplu de adăugare de grupă cu același nume:

The screenshot shows the 'Form1' application window. On the left, there is a table with columns 'Grupe', 'Sala', and 'Interval'. The 'Grupe' column has a blue header and a row with a double asterisk (**). An error dialog box titled 'Eroare' (Error) is displayed in the center, with a red 'X' icon and the message 'Grupe 1408A exista deja' (Group 1408A already exists). The dialog has an 'OK' button. In the background, the form contains two text areas: the left one lists '1408A (Intervale: (11, 13), (13, 15))', '1409A (Intervale: (13, 15), (15, 17))', and '1410A (Intervale: (15, 17), (17, 19))'; the right one contains 'C0-1 (Libera)'. Below these are input fields for 'Grupe' (containing '1408A') and 'Sala' (containing 'C0-1'), and buttons for 'Adauga sala' (Add room), 'Rezervare' (Reservation), 'Rezerva sala' (Reserve room), 'Sterge Grupe' (Delete groups), and 'Sterge Sali' (Delete rooms). At the bottom, there are buttons for 'Genereaza orar' (Generate schedule), 'Sterge orar' (Delete schedule), and 'Exemple' (Examples) with sub-buttons 'Exemplu 1', 'Exemplu 2', and 'Exemplu 3'.

Exemplu de adăugare de sală cu același nume:

The screenshot shows the 'Form1' application window. On the left, there is a table with columns 'Grupe', 'Sala', and 'Interval'. The 'Grupe' column has a blue header and a row with a double asterisk (**). An error dialog box titled 'Eroare' (Error) is displayed in the center, with a red 'X' icon and the message 'Sala C0-1 exista deja.' (Room C0-1 already exists). The dialog has an 'OK' button. In the background, the form contains two text areas: the left one lists '1408A (Intervale: (11, 13), (13, 15))', '1409A (Intervale: (13, 15), (15, 17))', and '1410A (Intervale: (15, 17), (17, 19))'; the right one contains 'C0-1 (Libera)'. Below these are input fields for 'Grupe' and 'Sala' (containing 'C0-1'), and buttons for 'Adauga sala' (Add room), 'Rezervare' (Reservation), 'Rezerva sala' (Reserve room), 'Sterge Grupe' (Delete groups), and 'Sterge Sali' (Delete rooms). At the bottom, there are buttons for 'Genereaza orar' (Generate schedule), 'Sterge orar' (Delete schedule), and 'Exemple' (Examples) with sub-buttons 'Exemplu 1', 'Exemplu 2', and 'Exemplu 3'.

6. Concluzii

Proiectul demonstrează utilizarea algoritmului Forward Checking pentru rezolvarea problemelor de alocare. Implementarea este eficientă și oferă flexibilitate utilizatorului prin adăugarea manuală de săli și grupe și prin posibilitatea de a genera și reseta orarul.

7. Bibliografie

1. Florin Leon, “Inteligență artificială, Curs 3, Jocuri. Satisfacerea constrângerilor”
2. Documentația oficială Microsoft C#