

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

**FACULTATEA DE INFORMATICĂ**



LUCRARE DE LICENȚĂ

**Implementarea unui algoritm de programare  
dinamică în Dafny ce rezolvă problema de  
selecție a activităților cu profit maxim**

propusă de

**Roxana Mihaela Timon**

**Sesiunea: februarie, 2024**

Coordonator științific

**Conf. Dr. Ciobâcă Ștefan**

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

**FACULTATEA DE INFORMATICĂ**

**Implementarea unui algoritm de  
programare dinamică în Dafny ce  
rezolvă problema de selecție a  
activităților cu profit maxim**

**Roxana Mihaela Timon**

**Sesiunea: februarie, 2024**

Coordonator științific

**Conf. Dr. Ciobâcă Ștefan**

Avizat,  
Îndrumător lucrare de licență,  
Conf. Dr. Ciobâcă Ștefan.

Data: ..... Semnătura: .....

### **Declarație privind originalitatea conținutului lucrării de licență**

Subsemnatul **Timon Roxana Mihaela** domiciliat în **România, jud. Vaslui, sat. Valea-Grecului, str. Bisericii, nr. 24**, născut la data de **09 iulie 2000**, identificat prin CNP **6000709375208**, absolvent al Facultății de informatică, **Facultatea de informatică** specializarea **informatică**, promoția 2022, declar pe propria răspundere cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art. 143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul **Implementarea unui algoritm de programare dinamică în Dafny ce rezolvă problema de selecție a activităților cu profit maxim** elaborată sub îndrumarea domnului **Conf. Dr. Ciobâcă Ștefan**, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului ei într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data: .....

Semnătura: .....

## **Declarație de consimțământ**

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Implementarea unui algoritm de programare dinamică în Dafny ce rezolvă problema de selecție a activităților cu profit maxim**, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test, etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de informatică.

De asemenea, sunt de acord ca Facultatea de informatică de la Universitatea "Alexandru-Ioan Cuza" din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent **Roxana Mihaela Timon**

Data: .....

Semnătura: .....

# Cuprins

<b>Motivație</b>	<b>2</b>
<b>Intenție</b>	<b>3</b>
<b>Introducere</b>	<b>4</b>
<b>1 Dafny</b>	<b>5</b>
1.1 Prezentare generală . . . . .	5
1.2 Tipuri de date . . . . .	5
1.3 Metode , funcții si predicate . . . . .	6
1.3.1 Metode . . . . .	6
1.3.2 Funcții . . . . .	7
1.3.3 Predicate . . . . .	7
1.4 Precondiții, poscondiții si invariante de buclă . . . . .	8
1.4.1 Precondiții, poscondiții . . . . .	8
1.4.2 Invarianti de buclă . . . . .	8
1.4.3 Terminarea buclei . . . . .	9
1.5 Aplicabilitate . . . . .	9
<b>2 Programarea dinamică</b>	<b>10</b>
2.1 Probleme rezolvate cu ajutorul programării dinamice . . . . .	10
2.2 Alte tehnici de proiectare a algoritmilor . . . . .	11
2.3 Avantajele programării dinamice față de celelalte tehnici de proiectare .	11
<b>3 Problema de selecție a activităților cu profit maxim</b>	<b>13</b>
3.1 Descrierea algoritmului . . . . .	13
3.2 Datele problemei . . . . .	14
3.2.1 Reprezentarea datelor de intrare si a celor de ieșire . . . . .	15

3.2.2	Tipuri de date folosite în dezvoltarea problemei . . . . .	15
3.3	Implementarea propriu-zisă . . . . .	15
3.3.1	Detalii de implementare . . . . .	15
3.3.2	Metode, funcții, lemme si predicate folosite . . . . .	15
3.3.3	Precondiții, postcondițiilor și invariante . . . . .	15
3.3.4	Timeout . . . . .	15
3.4	Mod de lucru . . . . .	15
3.4.1	Assume false . . . . .	15
	<b>Concluzii</b>	<b>16</b>
	<b>Bibliografie</b>	<b>17</b>

# Motivație

Am ales sa fac această temă deoarece Dafny era un limbaj de programare nou pentru mine si am considerat a fi o provocare. Știam doar că acesta este folosit pentru a asigura o mai mare siguranță si corectitudine, putând fi aplicat in industria aerospațiala, în industria medicala, la dezvoltarea sistemelor financiare, în securitate și criptografie. Totodată, prin demonstrarea de corectitudine a unui algoritm in Dafny puteam să-mi folosesc pe langă cunostințele informatice si pe cele de matematică, de care am fost mereu atrasă. Pentru a crește gradul de complexitate al lucrării am decis sa folosesc ca și tehnică de proiectare programarea dinamică. Astfel, având posibilitatea sa înțeleg mai bine cum funcționează programarea dinamică și să demonstrez că, cu ajutorul ei se obține o soluție optimă.

# Intenție

În cadrul lucrării voi discuta despre limbajul de programare Dafny, surprinzând particularitățile sale, despre problema de selecție a activităților cu profit maxim, despre programarea dinamică și avantajele sale în comparație cu alte tehnici de proiectare a algoritmilor. Voi prezenta, de asemenea, demonstrația de corectitudine a problemei de selecție a activităților cu profit maxim folosind programarea dinamică în Dafny.



# Introducere

Lucrarea este structură în 3 capitole:

- **Dafny.** În acest capitol voi prezenta particularitățile limbajului de programare Dafny și ariile de aplicabilitate.
- **Programarea dinamică.** În acest capitol voi reaminti despre programarea dinamică ca tehnică de proiectare a algoritmilor, despre avantajele sale în comparație cu alte tehnici de proiectare, precum Greedy.
- **Problema de selecție a activităților cu profit maxim.** Acest capitol conține prezentarea algoritmului de selecție a activităților cu profit maxim, demonstrația de corectitudine cu ajutorul limbajului de programare Dafny și tehnica de lucru abordată.

# Capitolul 1

## Dafny

### 1.1 Prezentare generală

Dafny este un limbaj imperativ de nivel înalt cu suport pentru programarea orientată pe obiecte. Metodele realizate în Dafny au precondiții, postcondiții și invarianți care sunt verificate la compilare, bazându-se pe soluționatorul SMT Z3. În cazul în care o postcondiție nu poate fi stabilită (fie din cauza unui timeout, fie din cauza faptului că aceasta nu este valabilă), compilarea eșuează. Prin urmare, putem avea un grad ridicat de încredere într-un program verificat cu ajutorul sistemului Dafny. Acesta a fost conceput pentru a facilita scrierea unui cod corect, în sensul de a nu avea erori de execuție, dar și corect în sensul de a face ceea ce programatorul a intenționat să facă.

### 1.2 Tipuri de date

În Dafny există mai multe tipuri de date care pot fi utilizate pentru a defini variabile și structuri de date:

- **int**: folosit pentru a declara numere întregi  
*ex. var x: int := 10;*
- **bool**: folosit pentru a declara valori de adevăr: true și false  
*ex. var esteAdevarat: bool := true;*
- **char**: folosit pentru a declara caractere  
*ex. var litera: char := 'A';*

- **string** : folosit pentru a declara siruri de caractere  
ex. *var mesaj: string := "Am terminat codul!"*;
- **seq<T>**: folosit pentru a declara secvente de elemente ordonate de tip T, precum liste, cozi si stive, fiind imutabile odata ce au fost create  
ex. *var oSeq: seq<int> := {1, 2, 3}*;
- **set<T>**: folosit pentru a declara seturi de elemente unice de tip T  
ex. *var unSet: set<int> := {1, 2, 3}*;

Acestea sunt doar cateva exemple, limbajul oferă suport si pentru structuri de date mai complexe, inclusiv tipuri de date definite de utilizator.

## 1.3 Metode , funcții si predicate

### 1.3.1 Metode

Metodele în Dafny reprezintă blocuri de instrucțiuni ce pot avea un comportament stabilit, apelate sau pentru a calcula valori, sau pentru a realiza anumite instrucțiuni sau ambele. Fiecare metoda poate fi caracterizată de precondiții si postcondiții care trebuie îndeplinite. Precondițiile trebuie să fie îndeplinite la apelarea metodei pentru ca aceasta să se execute, iar postcondițiile trebuie sa fie îndeplinite după ce se termină de executat corpul metodei. O metoda a caror postcondiții nu sunt îndeplinite nu poate fi demonstrată ca fiind corectă. Postcondițiile reprezinta ceea ce Dafny știe în urma apelului unei metode, el uitând corpul acesteia.

O metodă în Dafny este definită folosind cuvântul cheie '*method*'.

```
method SumaCuVerificareM(a: int, b: int) returns (rezultat: int)
    ensures rezultat == a + b
{
    return a + b;
}
```

Cu ajutorul postconditiei *ensures rezultat == a + b* assert-ul urmator va putea fi evaluat cu true. Altfel, dacă ensures-ul ar fi lipsit, rezultatul funcției, respectiv *AdunaCuVerificareM(3,2)* nu poate fi știut.

```

method Main()
{
    var suma := AdunaCuVerificareM(3,2);
    assert suma == 5;
}

```

### 1.3.2 Funcții

Funcțiile, în schimb contin o singura instrucțiune și au ca scop calcularea unor funcții pur matematice. Spre deosebire de metode, Dafny nu uită corpul unei funcții atunci când acestea sunt apelate din exterior: alte metode, lemme. Funcțiile nu fac niciodată parte din programul final compilat, ele sunt doar instrumente care ne ajută să ne verificăm codul. O funcție în Dafny este definită folosind cuvântul cheie **'function'**.

```

function AdunaCuVerificareF(a: int, b: int): int
{
    a + b;
}

```

În acest caz, dacă vom apela în *main()* funcția *AdunaCuVerificareF(3,2)* assert-ul va avea loc cu succes.

```

method Main()
{
    var suma := AdunaCuVerificareF(3,2);
    assert suma == 5;
}

```

### 1.3.3 Predicate

În Dafny, predicatele sunt utilizate pentru a specifica condiții sau proprietăți care trebuie să fie adevărate în anumite contexte, cum ar fi în invariantii buclelor sau postcondițiile metodelor. Predicatele sunt adesea exprimate ca expresii booleene și joacă un rol crucial în specificarea proprietăților de corectitudine. Predicatul de mai jos asigură că variabila *n* este număr par. Un predicat în Dafny este definit folosind cuvântul cheie **'predicate'**.

```

predicate EstePar(n: int)
{
    n % 2 == 0
}

```

## 1.4 Precondiții, poscondiții si invariianți de buclă

### 1.4.1 Precondiții, poscondiții

Precondițiile și postcondițiile reprezintă proprietăți care trebuie să fie îndeplinite la intrarea (precondiții) și respectiv, la ieșirea (postcondiții) dintr-o metodă, leamnă, funcție. Astfel, adevărata putere a Dafny-ului vine din capacitatea de a adnota metodele pentru a le specifica comportamentul. O precondiție în Dafny este definită folosind cuvântul cheie **'requires'**. O postcondiție în Dafny este definită folosind cuvântul cheie **'ensures'**.

```

method ElementNeutru(x: int, y: int) returns (suma: int)
    requires x == 0
    ensures suma == y
{
    returns x + y;
}

```

În metoda de mai sus, pentru ca suma să fie egală cu valoarea lui y trebuie ca x să respecte precondiția *requires x == 0*.

### 1.4.2 Invarianti de buclă

Un invariant de buclă este o expresie care se menține chiar înainte de testul buclei, adică la intrarea într-o buclă și după fiecare execuție a corpului buclei. Aceasta surprinde ceva care este invariabil, adică nu se schimbă, la fiecare pas al buclei.

```

var i := 0;
while i < n
    invariant 0 <= i
{

```

```

    i := i + 1;
}

```

Trebuie să demonstrăm că executarea corpului buclei încă o dată face ca invariantul să fie valabil din nou. La fel cum Dafny nu va descoperi singur proprietățile unei metode, nu va ști că se păstrează alte proprietăți decât cele elementare ale unei bucle, decât dacă i se spune prin intermediul unui invariant. Cu alte cuvinte, după executarea corpului buclei, Dafny va ști doar proprietățile declarate cu ajutorul invariantilor.

### 1.4.3 Terminarea buclei

Dafny trebuie să demonstreze că o buclă `while` nu se execută la nesfârșit, prin utilizarea adnotărilor *decreases*. Există două locuri în care Dafny trebuie să demonstreze terminarea: buclele și recursivitatea. Ambele situații necesită fie o adnotare explicită, fie o presupunere corectă din partea lui Dafny. Ca terminarea să poate fi demonstrată trebuie ca Dafny să verifice că expresia devine din ce în ce mai mică, și că aceasta are o limită inferioară.

## 1.5 Aplicabilitate

Asigurând încredere și corectitudine, Dafny poate fi folosit în numeroase industrii, precum:

- **industria aerospațială** (la navigație, comunicații și controlul bordului)
- **industria medicală** (dezvoltarea aparaturii medicale, garantând corectitudinea algoritmilor care recepționează și evaluează diferite semnale)
- **în criptografie** (la dezvoltarea algoritmilor care asigură confidențialitatea și integritatea datelor)
- **securitate cibernetică** (firewall-uri sau sisteme de detectare a intruziunilor, ar putea fi verificate în mod oficial pentru a se asigura că acestea identifică și răspund corect la amenințările de securitate fără a introduce vulnerabilități)
- **industria automobilelor autonome** (Dafny ar putea fi utilizat pentru a verifica corectitudinea algoritmilor de percepție, de luare a deciziilor și de control, reducând astfel riscul de accidente).

# Capitolul 2

## Programarea dinamică

Facilisi nullam vehicula ipsum a arcu. Purus semper eget dui at tellus at. Adipiscing tristique risus nec feugiat. Eu volutpat odio facilisis mauris sit. Porta nibh venenatis cras sed. Penatibus et magnis dis parturient. Sollicitudin aliquam ultrices sagittis orci a. Senectus et netus et malesuada fames ac turpis egestas integer. Cras tincidunt lobortis feugiat vivamus at augue eget arcu dictum. Leo vel fringilla est ullamcorper eget nulla facilisi etiam dignissim. Nulla aliquet enim tortor at auctor urna nunc id cursus. Elit dui tristique sollicitudin nibh. Sagittis nisl rhoncus mattis rhoncus urna neque viverra. Convallis posuere morbi leo urna molestie at. Quisque egestas diam in arcu cursus euismod.

### 2.1 Probleme rezolvate cu ajutorul programării dinamice

A diam sollicitudin tempor id eu nisl. Hac habitasse platea dictumst vestibulum. Integer enim neque volutpat ac tincidunt. Facilisi nullam vehicula ipsum a arcu cursus vitae congue. Vel turpis nunc eget lorem. Vestibulum mattis ullamcorper velit sed ullamcorper morbi tincidunt ornare. Nunc sed blandit libero volutpat. Sit amet luctus venenatis lectus magna fringilla urna porttitor. Hac habitasse platea dictumst quisque sagittis purus. Sed faucibus turpis in eu mi bibendum neque egestas. Vel orci porta non pulvinar neque laoreet suspendisse interdum consectetur. Erat nam at lectus urna dui convallis convallis tellus id. Tristique sollicitudin nibh sit amet commodo nulla facilisi nullam vehicula. Etiam dignissim diam quis enim lobortis scelerisque. Nunc congue nisi vitae suscipit tellus mauris a diam maecenas. Lacus viverra vitae congue eu consequat ac felis donec. Mauris sit amet massa vitae tortor condimentum. Mauris

augue neque gravida in. Lorem ipsum dolor sit amet. Arcu dui vivamus arcu felis bibendum ut tristique et.

## **2.2 Alte tehnici de proiectare a algoritmilor**

Sit amet mauris commodo quis imperdiet massa tincidunt nunc pulvinar. Ligula ullamcorper malesuada proin libero nunc consequat interdum. Mauris a diam maecenas sed enim ut. Ut sem nulla pharetra diam sit amet nisl suscipit adipiscing. Leo dui ut diam quam nulla. Neque ornare aenean euismod elementum. Vitae sapien pellentesque habitant morbi tristique senectus. Lectus magna fringilla urna porttitor rhoncus dolor purus non enim. Egestas sed sed risus pretium quam vulputate dignissim suspendisse in. At quis risus sed vulputate odio ut enim. Hac habitasse platea dictumst quisque sagittis. Lectus vestibulum mattis ullamcorper velit sed. Massa vitae tortor condimentum lacinia quis vel eros donec ac. Vulputate dignissim suspendisse in est ante. Sed faucibus turpis in eu mi bibendum neque. Enim eu turpis egestas pretium aenean pharetra magna. Tellus mauris a diam maecenas.

## **2.3 Avantajele programării dinamice față de celelalte tehnici de proiectare**

Faucibus ornare suspendisse sed nisi lacus sed. Mi in nulla posuere sollicitudin aliquam ultrices. Lacus suspendisse faucibus interdum posuere lorem ipsum dolor sit amet. Odio tempor orci dapibus ultrices in iaculis nunc sed augue. Congue eu consequat ac felis donec et odio. Enim ut sem viverra aliquet eget sit amet. Sit amet consectetur adipiscing elit dui tristique sollicitudin. Quis blandit turpis cursus in. Cras fermentum odio eu feugiat pretium nibh ipsum consequat nisl. Non curabitur gravida arcu ac tortor dignissim convallis aenean. Porta non pulvinar neque laoreet suspendisse interdum consectetur libero id. Lacus viverra vitae congue eu consequat ac felis. Vulputate dignissim suspendisse in est ante in nibh mauris. Amet mauris commodo quis imperdiet massa. Varius sit amet mattis vulputate enim nulla aliquet. Pellentesque diam volutpat commodo sed egestas egestas. Amet est placerat in egestas erat imperdiet sed euismod. Scelerisque varius morbi enim nunc faucibus a pellentesque sit. Ut sem viverra aliquet eget sit amet tellus cras. Sem integer vitae justo eget magna



fermentum iaculis eu.

## Capitolul 3

# Problema de selecție a activităților cu profit maxim

Amet venenatis urna cursus eget. Quam vulputate dignissim suspendisse in est ante. Proin nibh nisl condimentum id. Egestas maecenas pharetra convallis posuere morbi. Risus viverra adipiscing at in. Vulputate eu scelerisque felis imperdiet. Cras adipiscing enim eu turpis egestas pretium aenean pharetra. In aliquam sem fringilla ut morbi tincidunt augue. Montes nascetur ridiculus mus mauris. Viverra accumsan in nisl nisi scelerisque eu ultrices vitae. In nibh mauris cursus mattis molestie a iaculis. Interdum consectetur libero id faucibus nisl tincidunt eget. Gravida in fermentum et sollicitudin ac orci. Suscipit adipiscing bibendum est ultricies. Etiam non quam lacus suspendisse. Leo urna molestie at elementum eu facilisis sed odio morbi. Egestas congue quisque egestas diam in arcu cursus. Amet consectetur adipiscing elit ut aliquam purus.

### 3.1 Descrierea algoritmului

Eros donec ac odio tempor. Facilisi morbi tempus iaculis urna id volutpat. Faucibus in ornare quam viverra orci sagittis eu. Amet tellus cras adipiscing enim eu turpis egestas. Integer feugiat scelerisque varius morbi. Platea dictumst vestibulum rhoncus est pellentesque elit ullamcorper dignissim. Bibendum arcu vitae elementum curabitur. Eu nisl nunc mi ipsum faucibus. Id aliquet lectus proin nibh nisl condimentum id venenatis a. Cras adipiscing enim eu turpis egestas pretium. Quisque non tellus orci ac auctor augue mauris augue. Malesuada pellentesque elit eget gravida cum. Ut

lectus arcu bibendum at. Massa id neque aliquam vestibulum morbi blandit. Posuere ac ut consequat semper viverra nam. Viverra adipiscing at in tellus integer feugiat scelerisque varius morbi. Morbi enim nunc faucibus a pellentesque sit amet porttitor eget. Eu feugiat pretium nibh ipsum consequat nisl vel. Nisl purus in mollis nunc sed.

## 3.2 Datele problemei

Elementum sagittis vitae et leo duis ut diam quam nulla. Purus sit amet volutpat consequat mauris nunc. Tincidunt augue interdum velit euismod in pellentesque massa. Nunc sed augue lacus viverra vitae congue. Porttitor leo a diam sollicitudin. Faucibus pulvinar elementum integer enim. Adipiscing bibendum est ultricies integer quis auctor elit. Blandit aliquam etiam erat velit scelerisque in. A iaculis at erat pellentesque adipiscing commodo elit at. Erat nam at lectus urna duis. Consequat ac felis donec et. Fermentum posuere urna nec tincidunt praesent semper feugiat nibh sed. Proin gravida hendrerit lectus a. Pretium viverra suspendisse potenti nullam ac tortor vitae purus. Arcu cursus euismod quis viverra nibh cras pulvinar mattis. Gravida arcu ac tortor dignissim convallis aenean. Quam nulla porttitor massa id neque aliquam vestibulum morbi. Sed viverra ipsum nunc aliquet. Quis enim lobortis scelerisque fermentum dui faucibus in.

**3.2.1 Reprezentarea datelor de intrare si a celor de ieşire**

**3.2.2 Tipuri de date folosite în dezvoltarea problemei**

### **3.3 Implementarea propriu-zisă**

**3.3.1 Detalii de implementare**

**3.3.2 Metode, funcţii, lemme si predicate folosite**

**3.3.3 Precondiţii, postcondiţiilor şi invarianţi**

**3.3.4 Timeout**

### **3.4 Mod de lucru**

**3.4.1 Assume false**

# Concluzii

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nunc mattis enim ut tellus elementum sagittis vitae et. Placerat in egestas erat imperdiet sed euismod. Urna id volutpat lacus laoreet non curabitur gravida. Blandit turpis cursus in hac habitasse platea. Eget nunc lobortis mattis aliquam faucibus. Est pellentesque elit ullamcorper dignissim cras tincidunt lobortis feugiat. Viverra maecenas accumsan lacus vel facilisis volutpat est. Non odio euismod lacinia at quis risus sed vulputate odio. Consequat ac felis donec et odio pellentesque diam volutpat commodo. Etiam sit amet nisl purus in. Tortor condimentum lacinia quis vel eros donec. Phasellus egestas tellus rutrum tellus pellentesque eu tincidunt. Aliquam id diam maecenas ultricies mi eget mauris pharetra. Enim eu turpis egestas pretium.

# Bibliografie

- Author1, *Book1*, 2018
- Author2, *Boook2*, 2017
- <https://cgi.cse.unsw.edu.au/eptcs/paper.cgi?FROM2019.1.pdf>
- <https://dafny.org/dafny/toc>
- <https://www.geeksforgeeks.org/weighted-job-scheduling/>
- <https://sites.google.com/view/fii-pa/2022/lectures?authuser=0>
- <https://profs.info.uaic.ro/stefan.ciobaca/wollic2021slides.pdf>