Safety and Dependability

Group member

Chunyu Gong(201608534)

How-Hsuan Hung (201608540)

Kang Miao (201622448)

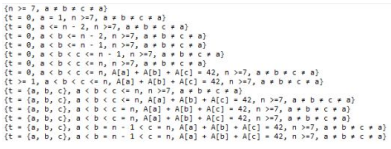
1. Determines the number of triples

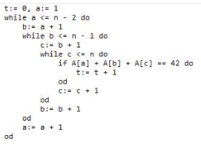
```
t:= 0, a:= 1
while a <= n - 2 do
    b:= a + 1
    while b <= n - 1 do
        c:= b + 1
        while c <= n do
            if A[a] + A[b] + A[c] == 42 do
                t:= t + 1
            od
            c:= c + 1
        od
        b:= b + 1
    od
    a:= a + 1
od
terminated
```

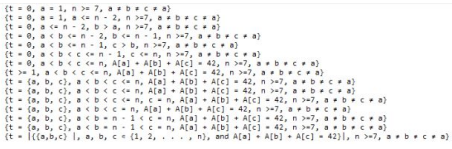
2. Prove its partial correctness

```
{n >= 7, a ≠ b ≠ c ≠ a}
                t:= 0, a:= 1
   {t = 0, a = 1, n >= 7, a ≠ b ≠ c ≠ a}
{t = 0, a = 1, n >=7, a ≠ b ≠ c ≠ a}
               while a <= n - 2 do
  {t = 0, a = 1, a <= n - 2, n >=7, a ≠ b ≠ c ≠ a}
{t = 0, a <= n - 2, n >=7, a ≠ b ≠ c ≠ a}
                    b:= a + 1
    {t = 0, a <= n - 2, b > a, n >=7, a ≠ b ≠ c ≠ a}
{t = 0, a < b <= n - 2, n >=7, a ≠ b ≠ c ≠ a}
                    while b <= n - 1 do
    {t = 0, a < b <= n - 2, b <= n - 1, n >=7, a ≠ b ≠ c ≠ a}
```

```
{t = 0, a < b <= n - 1, n >=7, a ≠ b ≠ c ≠ a}
                        c:= b + 1
    {t = 0, a < b <= n - 1, c > b, n >=7, a ≠ b ≠ c ≠ a}
{t = 0, a < b < c <= n - 1, n >=7, a ≠ b ≠ c ≠ a}
                        while c <= n do
    {t = 0, a < b < c <= n - 1, c <= n, n >=7, a ≠ b ≠ c ≠ a}
{t = 0, a < b < c <= n, n >=7, a ≠ b ≠ c ≠ a}
                            if A[a] + A[b] + A[c] == 42 do
   {t = 0, a < b < c <= n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠ a}
{t = 0, a < b < c <= n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠ a}
                                t:= t + 1
  {t >= 1, a < b < c <= n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠ a}
{t >= 1, a < b < c <= n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠ a}
                            od
   {t = {a, b, c}, a < b < c <= n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠
a}
{t = {a, b, c}, a < b < c <= n, n >=7, a ≠ b ≠ c ≠ a}
                            c:= c + 1
    {t = {a, b, c}, a < b < c <= n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠
a}
{t = {a, b, c}, a < b < c <= n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠ a}
                        od
   {t = {a, b, c}, a < b < c <= n, c = n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b
≠ c ≠ a}
{t = {a, b, c}, a < b < c = n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠ a}
                        b:= b + 1
    {t = {a, b, c}, a < b < c = n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠
a}
{t = {a, b, c}, a < b < c = n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c ≠ a}
                    od
   {t = {a, b, c}, a < b = n - 1 < c = n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b
≠ c ≠ a}
{t = {a, b, c}, a < b = n - 1 < c = n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c
≠ a}                a:= a + 1
  {t = {a, b, c}, a < b = n - 1 < c = n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠
c ≠ a}
{t = {a, b, c}, a < b = n - 1 < c = n, A[a] + A[b] + A[c] = 42, n >=7, a ≠ b ≠ c
≠ a}           od
   {t = |{{a,b,c} |, a, b, c ∈ {1, 2, . . . , n}, and A[a] + A[b] + A[c] = 42}|,
n >=7, a ≠ b ≠ c ≠ a}
```







3. Prove that it terminates

The program has three while loops from line 3,5,7. the first loop executes n-2-a times, a is the loop variable at this iteration, the second loop executes n-1-b times, the third loop which is the inner loop executes n - c times

Therefore, it has the same iteration counts as this

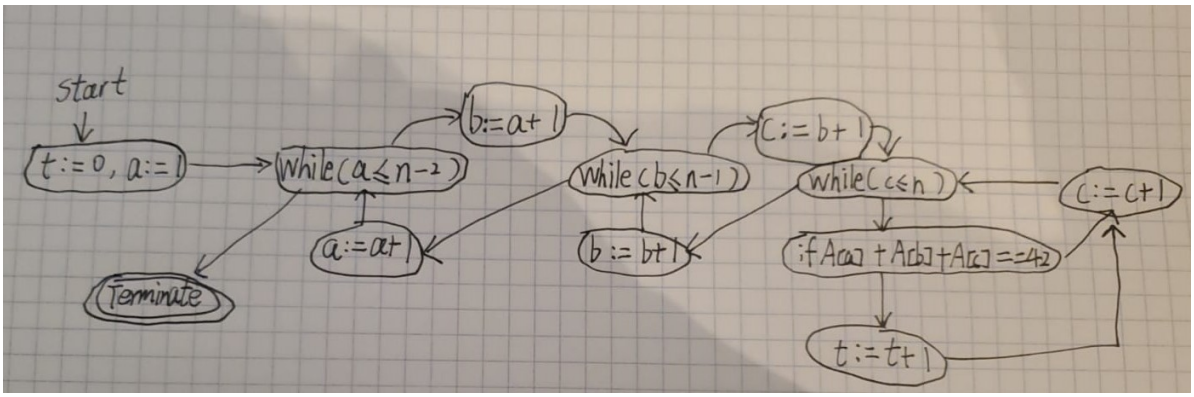
```
n = len(array)
for a in range(n-2):
    for b in range(a+1, n-1):
        for c in range(b+1, n):
            if (A[a] + A[b] + A[c]) == 42:
                t += 1
```

which gives us:

```
total number of iterations = ((n-2)*(n-1)*(n))/6
```

This means that the number of iterations depends on the size of the input list.

Since the program consists only of loops with a finite number of iterations, it terminates for any input list/array `A`, and termination is guaranteed.



Briefly describe a contemporary research problem

Verifying the functional correctness of deep learning models is a matter of contemporary research. The development of deep learning has a moderate impact on the fields of image recognition, speech recognition and natural language processing. However, deep learning models are often difficult to understand and predict outcomes, making deep learning validation techniques challenging due to the model's complex, non-linear nature, and lack of transparency about its inner workings.

Related Papers:

1. Meng, M. H., Bai, G., Teo, S. G., Hou, Z., Xiao, Y., Lin, Y., & Dong, J. S. (2022). Adversarial robustness of deep neural networks: A survey from a formal verification perspective. IEEE Transactions on Dependable and Secure Computing.
2. Xiang, W., Musau, P., Wild, A. A., Lopez, D. M., Hamilton, N., Yang, X., ... & Johnson, T. T. (2018). Verification for machine learning, autonomy, and neural networks survey. arXiv preprint arXiv:1810.01989.