# 다익스트라의 '프로그래밍의 수련(修練)': 열 번째, 몇 가지 작은 예제들의 정형적 처리 – II
# (Dijkstra's "A Discipline of Programming": The Tenth Lecture, The Formal Treatment of Some Small Examples – II)

김 도 형
성신여자대학교 컴퓨터정보학부

Do-Hyung Kim
School of CSE, Sungshin Women's University
E-mail: dkim@sungshin.ac.kr; URL: http://cs.sungshin.ac.kr/~dkim

요 약

이번 튜토리얼에서는 두 번째로 실제 문제를 다익스트라의 프로그래밍 방법론에 의해 다루는 예제를 보인다. 두 개의 문제를 사용할 것인데, 첫 번째는 정수 나눗셈의 나머지를 구하는 것이고, 두 번째는 네 개의 수를 정렬하는 것이다.

**[1]세 번째 예제.**

고정된 $a(a \geq 0)$와 $d(d > 0)$에 대해, 다음 의 조건 $R$을 성립시킬 것을 요구받았다고 하자:

$$0 \leq r < d \text{ and } d \mid (a - r)$$

(여기에서 수직 막대 '|'는 '~는 ~의 약수이다' 라고 읽는다.) 달리 말해서, $a$를 $d$로 나눈 뒤 남는 음이 아닌 최소의 나머지 $r$을 계산해 달라고 요청 받은 것이다. 이 문제가 문제로서의 의미가 있게 하기 위하여, 우리는 이용하는 산술 연산들을 덧셈과 뺄셈만으로 제한한다. 항 $d \mid (a - r)$은 $r \boldsymbol{=} a$에 의해 만족되는데, $a \geq 0$이기 때문에 이것은 또한 $0 \leq r$도 만족하는 하나의 초기화가 되므로, 불변 관계 $P$로서 다음을 선택하자고 제안해 본다:

$$0 \leq r \text{ and } d \mid (a - r)$$

우리는 그것의 감소가 종료를 보장해야 하는 함수 $t$를 위해서는 $r$ 자신을 선택한다. $r$의 조작은 관계 $d \mid (a - r)$이 불변으로 유지되도록 하

1) 이번 튜토리얼은 다익스트라의 원저[1]에서 제8장의 일부에 해당하는 부분이다.

는 것이어야만 하므로, $r$은 오직 $d$의 배수만큼만 변경될 수 있을 것이다. 예를 들면 $d$ 자신만큼 말이다. 그래서 우리는 다음을 계산해 보도록 유도된다:

wp("$r := r - d$", $P$) and wdec("$r := r - d$", $r$) $=$
$0 \le r - d$ and $d \mid (a - r + d)$ and $d > 0$

항 $d > 0$은 불변 관계 $P$에 포함될 수도 있었으므로, 그렇다면 첫 번째 항만 함축되지 않는 셈이다;[2] 우리는 상응하는 가드 "$r \ge d$"와 초안 프로그램을 찾은 것이다:

```
if a ≥ 0 and d > 0 →
    r := a;
    do r ≥ d → r := r − d od
fi
```

마치면서 $P$ and non $r \ge d$의 참이 성립되며, 이것은 $R$을 함축하는 관계이고 따라서 문제는 풀린 것이다.

이제 추가적으로 $q$에 값을 배정하여 궁극적으로는 또한 다음이 성립하도록 만드는 것이 요구된다고 가정하자:

$$a = d * q + r$$

달리 말해서 몫 역시 계산하는 일을 요청 받은 것이다. 그러면 우리는 이 항을 불변 관계에 추가하는 것을 시도해 볼 수 있다. 다음 식

$$(a = d * q + r) \Rightarrow (a = d * (q + 1) + (r - d))$$

이 성립하므로, 우리는 다음 프로그램에 도달한다:

```
if a ≥ 0 and d > 0 →
    q, r := 0, q;
    do r ≥ d → q, r := q + 1, r − d od
fi
```

물론 위의 프로그램은 몫이 크다면 시간이 매우 오래 걸린다. 속도를 올릴 수 있을까? 그러기 위한 명백한 방법은 $r$을 $d$보다 큰 $d$의 배수만큼 감소시키는 것이다. 이럴 목적으로 새로운 변수, 예컨대 $dd$를 도입하게 되면, 성립시켜 불변으로 유지해야 할 관계는 다음과 같다:

$$d \mid dd \text{ and } dd \ge d$$

처음에 $d$였던 $dd$가 다소 빠르게 커지는 것, 예를 들면 매번 두 배가 되는 것을 허용하면서 "$r := r - d$"를 $r$에서 $dd$를 거듭 뺄 수 있는 것으로 대치함으로써, 우리 첫 번째 프로그램의 속도를 올릴 수 있다. 그래서 우리는 다음 프로그램을 고려하게 된다:

```
if a ≥ 0 and d > 0 →
  r := a;
  do r ≥ d →
    dd := d;
    do r ≥ dd → r := r − dd; dd := dd + dd od
  od
fi
```

명백히 관계 $0 \le r$ and $d \mid (a - r)$은 불변으로 유지되며, 따라서 이 프로그램은 만약 적절히 종료한다면 $R$을 성립시킨다. 그런데 적절히 종료하는가? 물론 그러하다. 왜냐하면 $dd > 0$이라서 종료하게 되는 내부 루프는 오직 $r \ge dd$를 만족하는 초기 상태에서만 활성화되고, 따라서 $r := r - dd$ 감소는 외부 루프가 매번 반복될 때마다 적어도 한 번 수행되기 때문이다.

---

2) 첫 번째 항 $0 \le r - d$만이 함축되지 않는 것이다. 두 번째 항 $d \mid (a - r + d)$는 불변 관계 $P$의 두 번째 항 $d \mid (a - r)$로부터 함축된다.

그런데 위와 같은 추론—비록 충분히 설득력이 있으나—은 매우 비정형적이며, 이 장을 ‘정형적 처리’라고 부르는 만큼, 우리가 직관에 의존했던 정리를 명확하게 하고 증명하는 일을 시도해 볼 수 있겠다.

IF와 DO 그리고 $BB$의 의미는 그대로 하고, P를 불변으로 유지되는 관계라고 하자. 즉, 다음이 성립한다:

$$\text{모든 상태에 대해, } (P \text{ and } BB) \Rightarrow \mathrm{wp}(\mathrm{IF},\ P) \tag{1}$$

그리고 $t$를 $t0$의 임의의 값과 모든 상태에 대해 다음이 성립하는 정수 함수라고 하자:

$$(P \text{ and } BB \text{ and } t \le t0 + 1) \Rightarrow \mathrm{wp}(\mathrm{IF},\ t \le t0) \tag{2}$$

또는 동등한 식으로서,

$$\text{모든 상태에 대해, } (P \text{ and } BB) \Rightarrow \mathrm{wdec}(\mathrm{IF},\ t) \tag{3}$$

그러면 $t0$의 임의의 값과 모든 상태에 대해 다음이 성립한다:

$$\begin{aligned}&(P \text{ and } BB \text{ and } \mathrm{wp}(\mathrm{DO},\ T) \text{ and } t \le t0 + 1)\\ &\quad\Rightarrow \mathrm{wp}(\mathrm{DO},\ t \le t0)\end{aligned} \tag{4}$$

또는 동등한 식으로서,

$$(P \text{ and } BB \text{ and } \mathrm{wp}(\mathrm{DO},\ T)) \Rightarrow \mathrm{wdec}(\mathrm{DO},\ t) \tag{5}$$

말로 하자면: 만일 선택된 각각의 가드 명령이 $t$의 실질적 감소를 일으킨다는 것을 불변으로 유지되는 관계 $P$가 보장한다면, 가드 명령이 적어도 한 번 실행된 후 적절히 종료하는 경우 그 반복 구조는 $t$의 실질적 감소를 일으킬 것이다. 이 정리는 매우 명백하므로 증명하기가 어려웠다면 유감이었을 터인데, 다행히 그렇지 않다. 우리는 (1)과 (2)로부터 임의의 값 t0과 모든 상태에 대해, 다음 식이 모든 k ≥ 0에 대해 성립한다는 것이 도출됨을 보인다:

$$\begin{aligned}&(P \text{ and } BB \text{ and } H_k(T) \text{ and } t \le t0 + 1)\\ &\quad\Rightarrow H_k(t \le t0)\end{aligned} \tag{6}$$

$(BB \text{ and } H_0(T)) = F$이므로 이 식은 $k = 0$에 대해 성립하며, 우리는 (6)이 $k = K$에 대해 성립한다는 가정으로부터 그것이 $k = K + 1$에 대해서도 마찬가지로 성립함을 유도해야 한다.

$$\begin{aligned}&(P \text{ and } BB \text{ and } H_{K+1}(T) \text{ and } t \le t0 + 1)\\ &\Rightarrow \mathrm{wp}(\mathrm{IF},\ P) \text{ and } \mathrm{wp}(\mathrm{IF},\ H_K(T)) \text{ and } \mathrm{wp}(\mathrm{IF},\ t \le t0)\\ &= \mathrm{wp}(\mathrm{IF},\ P \text{ and } H_K(T) \text{ and } t \le t0)\\ &\Rightarrow \mathrm{wp}(\mathrm{IF},\ (P \text{ and } BB \text{ and } H_K(T) \text{ and } t \le t0 + 1)\\ &\qquad \text{or } (t \le t0 \text{ and non } BB))\\ &\Rightarrow \mathrm{wp}(\mathrm{IF},\ H_K(t \le t0) \text{ or } \mathrm{H}_0(t \le t0))\\ &= \mathrm{wp}(\mathrm{IF},\ H_K(t \le t0))\\ &\Rightarrow \mathrm{wp}(\mathrm{IF},\ H_K(t \le t0)) \text{ or } \mathrm{H}_0(t \le t0)\\ &= H_{K+1}(t \le t0)\end{aligned}$$

첫 번째 함축은 (1)과 $H_{K+1}(T)$의 정의 그리고 (2)로부터 나온다; 세 번째 줄의 등가는 명백하다; 네 번째 줄의 함축은 ($BB$ or non $BB$)와 공접(conjunction)을 취한 뒤 두 항을 모두 약화시킴으로써 유도된다; 다섯 번째 줄의 함축은 $k = K$인 (6)과 $H_0(t \le t0)$의 정의로부터 나온다; 나머지는 뻔하다. 따라서 관계 (6)이 모든 $k \ge 0$에 대해 증명되었고, 이 결과로부터 (4)와 (5)는 바로 도출된다.

## 연 습 문 제

두 번째 프로그램 몫도 역시 계산하도록 수정하고, 그 프로그램에 대한 정형적 정확성 증명을 하라. (연습문제의 끝.)

다음은 작은 수, 예컨대 3이 있어서 우리가 그

것으로 곱하거나 나눌 수 있는데, 이 연산들은 매우 빨라서 사용하기에 좋다고 가정하자. 이 곱한 결과를 '$m * 3$'–또는 '$3 * m$'–으로, 몫을 '$m / 3$'으로 표기할 것이다; 후자의 식은 애초에 $3 \mid m$이 성립하는 경우에만 평가(evaluation)[3]가 요구될 것이다. (우리는 정수를 가지고 작업중이니까.)

우리는 또다시 원하는 관계 $R$을 반복 구조에 의해 성립시키려고 하는데, 이 관계에 대한 불변 관계 $P$는 상수를 변수로 대치함으로써 유도된다. 상수 d를 그 값이 $d *$ (3의 거듭제곱)으로 제한될 변수 $dd$로 대치하면, 우리는 다음 불변 관계 $P$에 이른다:

$$0 \le r < dd \text{ and } dd \mid (a - r) \text{ and } (\mathrm{E}\ i\text{: } i \ge 0\text{: } dd = d * 3^i)$$

이 관계를 확립하고, 그런 뒤 그것을 불변으로 유지하면서 $d = dd$를 만족하는 상태에 도달하려고 할 것이다.

그것을 확립하기 위해서는 추가의 반복 구조를 필요로 한다: 먼저 다음 관계를 확립한다:

$$0 \le r \text{ and } dd \mid (a - r) \text{ and } (\mathrm{E}\ i\text{: } i \ge 0\text{: } dd = d * 3^i)$$

그런 뒤 $dd$가 충분히 커져서 $r < dd$도 또한 만족될 때까지 증가시킨다. 다음 프로그램은 그런 일을 할 것이다:

```
if a ≥ 0 and d > 0 →
  r, dd := a, d;
  do r ≥ dd → dd := dd * 3 od;
  do dd ≠ d → dd := dd / 3;
                do r ≥ dd → r := r − dd od
  od
fi
```

3) 식 또는 표현(expression)의 값을 구하는 일을 '평가'라고 부른다.

**연 습 문 제**

위의 프로그램이 몫도 계산하도록 역시 수정하고, 그 프로그램에 대한 정형적 정확성 증명을 하라. $dd / 3$이 계산될 때는 언제나 원래 $3 \mid dd$가 성립한다는 점을 이 증명은 보여야 한다. (연습문제의 끝.)

위의 프로그램은 매우 흔한 특징을 보여준다. 바깥 레벨에서 우리는 두 개의 반복 구조를 연달아 가지고 있다; 우리가 두 개 이상의 반복 구조를 같은 레벨에서 연달아 가질 때, 뒤에 오는 것들의 가드 명령이 앞에 오는 것들의 가드 명령보다 더 정교한 경향이 있다. (이것은 '다익스트라의 법칙(Dijkstra's Law)'으로 알려져 있는데, 항상 성립하지는 않는다.) 이러한 경향의 이유는 명백하다: 각 반복 구조는 그것이 불변으로 유지한 관계에 각각의 'and non $BB$'를 추가하고, 이 추가적 관계는 다음 반복 구조에 의해 역시 불변으로 유지되어야 한다. 그런데 내부 루프를 위해, (위의 프로그램의) 두 번째 반복 구조는 첫 번째 것의 정확히 반대이다; 그러나 잠재적으로 깨어진 관계 r < dd, 즉 첫 번째 루프가 달성한 것을 회복하는 일이 바로 추가된 다음 문장의 기능이다:

```
do r ≥ dd → r := r − dd od
```

**네 번째 예제.**

고정된 $Q1, Q2, Q3, Q4$에 대해서, $R1$ and $R2$로 주어진 $R$을 성립시킬 것을 요구받았다. 여기서 $R1$과 $R2$는 다음과 같다:

$R1$: 값들의 연속 $(q1, q2, q3, q4)$는 값들의 연속 $(Q1, Q2, Q3, Q4)$의 한

순열(permutation)이다.

$R2$: $q1 \le q2 \le q3 \le q4$

$R1$을 불변으로 유지되어야 할 관계 $P$로 취한 한 가지 해는 다음과 같다:

```
q1, q2, q3, q4 := Q1, Q2, Q3, Q4;
do q1 > q2 → q1, q2 := q2, q1
 | q2 > q3 → q2, q3 := q3, q2
 | q3 > q4 → q3, q4 := q4, q3
od
```

첫 번째 치환문은 명백히 $P$를 확립하며 어떤 가드 명령도 그것을 깨뜨리지 않는다. 종료한다면 non $BB$가 성립하고, 그것이 관계 $R2$이다. 사람들이 이것이 정말 종료한다는 점을 스스로 확신하는 방식은 주로 그들의 배경에 달려있다: 수학자라면 반전(inversion)의 개수가 감소함을 관찰할 지도 모르겠고, 오퍼레이션 리서처(operations researcher)라면 그것을 $q1 + 2 * q2 + 3 * q3 + 4 * q4$를 최대로 만드는 것으로 해석할 것이며, 물리학자로서의 다익스트라는 단순히 무게 중심이 한쪽 방향(확실하게 하자면 오른쪽)으로 움직이는 것을 ‘본다’. 이 프로그램은 우리가 가드로서 무엇을 골랐던 간에 결코 관계 $P$를 깨뜨릴 위험은 없을 것이라는 뜻에서 주목할 만하다: 이 예제에서의 가드들은 순수히 종료라는 요구 사항의 결과인 것이다.

**주의.** 우리가 다음과 같은

```
q1 > q2 → q1, q2 := q2, q1
```

다른 대안들 역시 추가할 수도 있었음을 살펴보라; (그러나) 주어진 세 개들 중 하나를 대치하는 데 사용할 수는 없다. (**주의의 끝.**)

이것은 비결정성이 달성을 가능하게 만든 종류의 명료함의 좋은 예제이다; 그렇지만 말할 필요도 없이, 많은 개수의 값들을 이것과 유사한 방식으로 정렬하는 것은 추천하지 않는다.

## 참 고 문 헌

[1] Dijkstra, E. W., *A Discipline of Programming*, Prentice Hall, Englewood Cliffs, NJ, 1976.

김 도 형

1981 ~ 1985 서울대학교공과대학 컴퓨터공학과(학사)
1985 ~ 1987 한국과학기술원 전산학과(석사)
1987 ~ 1992 한국과학기술원 전산학과(박사)
1992 한국과학기술원 정보전자연구소 연수연구원
1992 ~ 현재 성신여자대학교 컴퓨터정보학부 교수
1997 ~ 1998 스토니 브룩 소재 뉴욕주립대학교 컴퓨터과학과 객원교수

관심분야 : 프로그래밍 언어