

조종관계에 의한 원천코드의 중간표시방법

최영화, 한도욱

위대한 령도자 김정일동지께서는 다음과 같이 교시하시였다.

《과학기술의 발전과 기술개조의 견지에서 볼 때 현시대는 자동화의 시대라고 말할수 있습니다. 지금 세계 여러 나라들에서 현대과학기술발전의 성과에 의거하여 생산을 자동화, 원격조종화하는 방향으로 나아가고있습니다.》(《김정일선집》 증보판 제4권 475페이지)

임베디드응용프로그램의 급속한 발전과 고성능체계구조들이 끊임없이 새롭게 갱신되고 높은 품질의 번역기들이 급속히 발전함에 따라 새로운 요구들이 계속 제기된다.

원천코드에 의한 추상중간표시방법은 프로그램의 이식성과 프로그램작성의 효과성 등 여러 측면에서 관건적인 작용을 한다.

선행연구[4]에서는 번역기의 처리용에서 효과적인 추상문장론서술에 기초한 중간표시방법을 제기하였으며 선행연구[1, 2]에서는 프로그램의 수학적모형과 여러가지 규칙들을 제기하였다. 또한 선행연구[3]에서는 평가규칙에 대한 해석적검증을 제기하였다.

논문에서는 원천코드의 이식성과 코드생성의 유효성 등의 측면에서 효과적인 작용을 하는 추상중간표시에 대한 대수적자료형의 표시방법을 제기함으로써 임베디드응용프로그램의 이식성에 효과적으로 리용할수 있게 하였지만 번역기의 앞단을 걸친 추상중간코드를 리용하기때문에 원천코드단계에서의 이식과 코드생성을 합리적으로 할수 없는 부족점이 있다.

여기서는 선행연구[4]에서의 중간표시방법을 확장하여 고급언어로 작성한 원천코드의 중간표시에 편리한 조종관계를 정의하고 중간표시방법과 그것에 필요한 수학적개념과 규칙들을 제기하였다.

1. 조종관계와 조종구조의 개념

프로그램을 수학적으로 모형화하고 수학적처리방법에 의한 성질을 연구하는것은 프로그램의 연구와 작성에서 수학적수단들을 리용하며 과학성을 담보하기 위한 중요한 요구이다.

먼저 프로그램을 수학적으로 모형화하기 위한 개념들을 정의한다.

대다수 프로그램작성언어들에서 프로그램은 순차, 가지, 반복, 함수호출, 재귀, 병행과 중단 등의 구조들이 공통적으로 리용된다.[2]


기초조종구조(BCS-Basic Control Structure)는 프로그램을 구성하는 구조들이다.

기초조종구조들의 기호와 분류는 표 1과 같다.

표 1의 기초조종구조는 선행연구[2]에서 제기한 구조에 정의구조를 보충한것이다.

$$BCS=\{ @, \langle, \perp, R^i, R^+, R^*, \xrightarrow{f}, \bigcirc, \swarrow, \parallel, \square \}$$

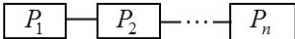
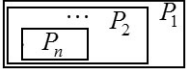
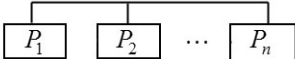
표 1. 기초조종구조(BCS)

No.	종류	BCS	기호표식
1	단순구조	sequence	@
2	가지구조	if-then-else case for-do	\langle \perp R^i
3	반복구조	repeat-until while-do	R^+ R^*
4	함수호출구조	function-call	\xrightarrow{f}
5	재귀구조	recursion	
6	병행구조	parallel	\parallel
7	중단구조	interrupt	⚡
8	정의구조	define	\square

기초조종구조들을 리용하여 복잡한 구조들을 구성하기 위한 $R_b = \{\rightarrow, \subset, \leftrightarrow\}$ 의 관계들의 모임을 조종관계모임이라고 부른다. 여기서 $R_1 : \rightarrow$ 은 순차관계, $R_2 : \supset$ 는 포함관계, $R_3 : \leftrightarrow$ 은 병렬관계이다.

관계들의 정의모형은 표 2와 같다.

표 2. 관계들의 정의모형

조종관계	표기	모형	기호표식
순차관계	\rightarrow		$P_1 \rightarrow P_2 \rightarrow \dots \rightarrow P_n$
포함관계	\supset		$P_1 \supset P_2 \supset \dots \supset P_n$
병렬관계	\leftrightarrow		$P_1 \leftrightarrow P_2 \leftrightarrow \dots \leftrightarrow P_n$

순차관계는 조종구조들사이의 순차적인 실행관계를 나타내는 관계로서 코드작성에서 같은 준위의 구조들의 관계를 보여준다. 포함관계는 한 조종구조가 다른 조종구조를 포함하고있는 관계를 나타낸다. 병렬관계는 조종구조들의 모임으로 이루어진 구조들사이의 관계로서 프로그램부분들의 서로 독립적인 관계를 나타낸다.

조종구조 T 는 n 개 기초조종구조들을 조종관계에 의하여 결합한 유한렬이다. 즉 $T = \mathbf{R}(S_i r_{ij} S_j) = (\dots(((S_1) r_{12} S_2) r_{23} S_3) \dots r_{n-1, n} S_n)$ 이다. 여기서 S_i, S_j 는 조종구조이고 $r_{ij} \in R_b$, $j=i+1$ 이며 n 은 조종구조 T 를 이루는 기초조종구조개수이다.

조종구조 T 는 기초조종구조들의 매물관계실체라는것을 알수 있다.

이제 조종구조모임을 생각하자.

조종구조모임 S 는 m 개의 조종구조들의 유한모임이라고 하자.

$$S = \bigcup_{k=1}^m T_k = \bigcup_{k=1}^m \left[\mathbf{R}_{i=1}^{n_k-1} (S_i^k r_{ij}^k S_j^k) \right] = \bigcup_{k=1}^m [(\dots(((S_1^k) r_{12}^k S_2^k) r_{23}^k S_3^k) \dots r_{n-1, n}^k S_n^k)], j=i+1$$

일반적으로 프로그램은 조종관계모임과 조종구조모임에 의해 표현될수 있으므로 프로그램은 조종구조들의 모임으로 된다.

2. 원천코드의 중간표시방법

추상중간표시(AIR)는 원천프로그램의 개별적인 언어특징에 무관계하게 표시하며 프로그램에서 조종구조들을 분리하거나 삽입하기 편리하도록 구성한다.

AIR는 조종구조들에 대수적산법을 적용할수 있도록 자료선언부와 본체로 구성한다.

AIR에서 자료선언부는 분리되는 조종구조의 의미가 보존되도록 대역적인 선언부로 구성하며 본체는 연산부로 구성한다. 본체에는 국부적인 선언부를 포함할수 있다.

아래에 AIR의 일부 규칙들을 BNF표기로 제시한다.

```
<Alg>→<def> | <body> | <def><Alg>, <def>→<block> | <function> | <file> | <class> | ∅
<block>→<private def> | <op body>, <private def>→<macro> | <bs>
<bs>→<int> | <char> | <real> | <string> | <label> | ∅, <op body>→<bs> | <bs><op body>
<bs>→@ | < | ⊥ |  $R^i$  |  $R^+$  |  $R^*$  |  $\xrightarrow{f}$  | || | < | □
```

프로그램부분이 다음과 같은 실례를 보기로 하자.

```
#define plus(a, b) a+b                A
```

```
int a=1;
```

```
int b=2;
```

```
int func(int x, int y)
```

```
{int c=plus(a, b)*3;
```

```
...
```

<pre>{int b=3; int d=plus(a, b); return d; }</pre>	B
--	---

```
...
```

```
return c;
```

```
}
```

블록 B를 분리하기 위한 AIR는 다음과 같다.

```
#define plus(a, b) a+b                //<def>
```

```
int a=1;
```

```
{int b=3;                            //<body>
```

```
int d=plus(a, b);
```

```
    return d;
```

```
}
```

```
//<def>
```

```
<define>< sequence>
```

```
//<body>
```

```
<Sequence>< sequence>< sequence>
```

실례에서 보여주는바와 같이 블록 B를 분리할 때 선언부 <def>는 매크로와 옹근수 형변수 a의 선언부로 구성되며 변수 a에 값 1을 부과하는 기초조종구조로 이루어진다.

연산부는 국부적인 용근수형변수 b 와 d 를 선언하며 3개의 기초조종구조를 포함하는 조종구조를 포함한다.

AIR는 다음과 같은 방법으로 한다.

① 원천프로그램에서 추상중간표시를 진행할 부분을 분리한다. 그 부분을 R 라고 표시한다.

② R 에서 리용되는 모든 변수, 함수, 매크로와 클래스들에서 대역적인것과 국부적으로 선언된것을 구분한다.

대역적으로 선언된 부분은 R 에 대역적선언부를 첨부한다.

국부적으로 선언된 부분에서 R 에서 리용되지 않은 변수들은 제거한다.

③ R 를 BCS표기로 중간표시를 진행한다.

3. 조종관계의 수학적규칙

기호조종구조, 조종구조들이 프로그램을 이루는 구조들이므로 이 구조들로 프로그램을 이식하거나 조립할 때 쓰이는 수학적규칙들을 보기로 하자.

R_1, R_2, R_3 을 조종구조라고 할 때 선행연구[1]에서 제기한 6가지 관계규칙중에서 순차, 포함, 병렬관계에서 리용되는 다음의 4가지 관계를 보기로 하자.

① 묶음법칙(Associative)

순차관계에 대하여 묶음법칙이 성립된다는것을 증명하자.

$$R_1 \rightarrow (R_2 \rightarrow R_3) = (R_1 \rightarrow R_2) \rightarrow R_3$$

웃식의 왼변은 조종구조 R_1, R_2, R_3 의 순차관계렬로서 프로그램구조 $R_1 \rightarrow R_2 \rightarrow R_3$ 과 같다. 오른변도 $R_1 \rightarrow R_2 \rightarrow R_3$ 과 구조적으로 같다.

그러므로 순차관계에 대하여 묶음법칙이 성립된다.

포함관계와 병렬관계도 유사하게 설명할수 있다.

② 반사성(Reflexive)

$R_1 \rightarrow R_2$ 와 $R_1 \leftrightarrow R_1$ 은 프로그램구조에서 성립되는 구조이므로 순차관계와 병렬관계에 대하여 반사성이 성립된다.

$R_1 \supset R_1$ 은 프로그램구조에서 늘 성립되지 않으므로 포함관계에 대하여 반사성이 성립되지 않는다.

③ 대칭성(Symmetric)

$R_1 \rightarrow R_2$ 인 R_1 과 R_2 에 대하여 프로그램구조에서 $R_2 \rightarrow R_1$ 은 성립되지 않으므로 포함관계에 대하여 대칭성이 성립되지 않는다.

$R_1 \supset R_2$ 인 R_1 과 R_2 에 대하여 $R_2 \supset R_1$ 은 일반적으로 성립되지 않으므로 포함관계에 대하여 대칭성이 성립되지 않는다.

$R_1 \leftrightarrow R_2$ 인 R_1 과 R_2 에 대하여 프로그램구조에서 $R_2 \leftrightarrow R_1$ 은 언제나 성립되므로 병렬관계에 대하여 대칭성이 성립된다.

④ 이행성(Transitive)

$R_1 \rightarrow R_2, R_2 \rightarrow R_3$ 으로 되는 R_2 가 있으나 프로그램구조에서 $R_1 \rightarrow R_3$ 이 성립되지 않

으므로 순차관계에 대하여 이행성이 성립되지 않는다.

$R_1 \supset R_2$, $R_2 \supset R_3$ 인 R_2 가 있으나 $R_1 \supset R_3$ 은 일반적으로 성립되지 않으므로 포함관계에 대하여 이행성이 성립되지 않는다.

$R_1 \leftrightarrow R_2$, $R_2 \leftrightarrow R_3$ 인 R_2 가 있고 $R_1 \leftrightarrow R_3$ 이 성립되므로 병렬관계에 대하여 이행성이 성립된다.

이상의 조종관계에 대한 구성관계규칙들의 성립관계는 표 3과 같다.

표 3. 구성관계규칙들의 성립관계

	표기	구성관계 규칙들			
		R_1	R_2	R_3	R_4
순차	\rightarrow	/			/
포함	\subset	/	/		/
병렬	\leftrightarrow	/		/	

표 3에서 /는 성립된다는것을 표시한다.

이상과 같이 프로그램을 조종구조들의 모임으로 추상중간표시를 진행하고 수학적 규칙들로 정의함으로써 프로그램의 재리용에 대수적모임과 관계산법을 적용할수 있다.

참 고 문 헌

- [1] Ying Xiu Wang; IEEE CCECE/CCGEI, 5, 2381, 2006.
- [2] Ying Xiu Wang; 5th IEEE Int. Conf. on Cognitive Informatics(Iccro6), 226, 2006.
- [3] Ron Lin et al.; SYM1/I-74899-1509, @ Soft COM, 2011.
- [4] 戴桂兰 等; 清华大学学报, 4, 499, 2003.

주체104(2015)년 6월 5일 원고접수

A Method for ATR of the Source Code by using the Control Relation

Choe Yong Hwa, Han To Uk

We defined the control relation which is convenient for ATR of the source code encoded by the high level language using the control relation, and we studied the mathematical rules for it.

Key word: control relation