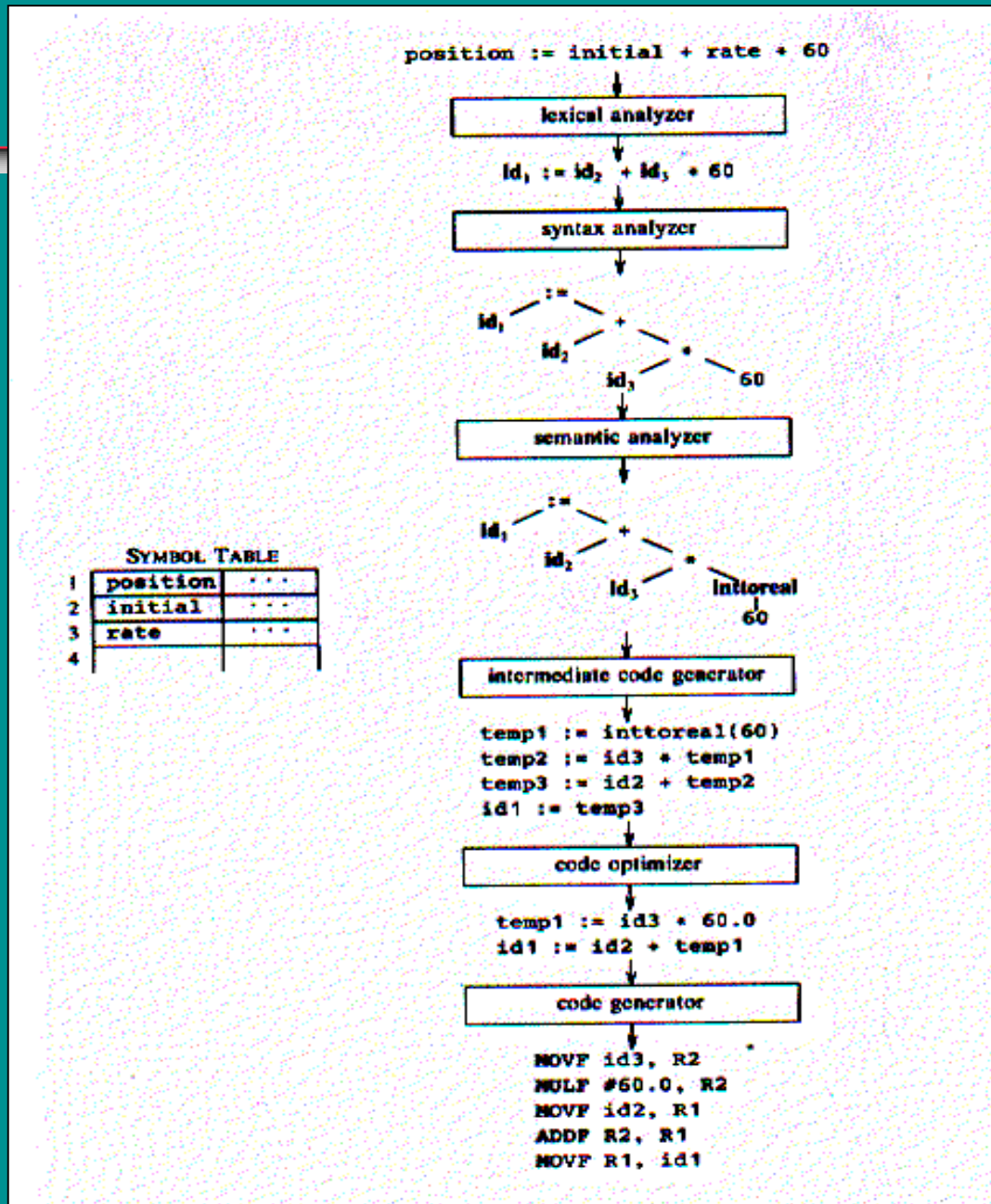


Esempio tratto da:

A.V. Aho, R. Sethi and J.D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1988



Queste fasi possono operare in pipelining

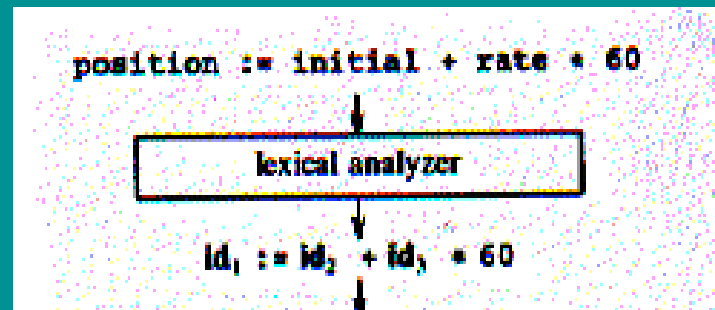
Esempio tratto da:

A.V. Aho, R. Sethi and J.D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1988

Raccoglie informazioni sui simboli
Introdotti e restituisce entry, id_i

SYMBOL TABLE

| | | |
|---|----------|-----|
| 1 | position | ... |
| 2 | initial | ... |
| 3 | rate | ... |
| 4 | | |



position := initial +

Un carattere per volta

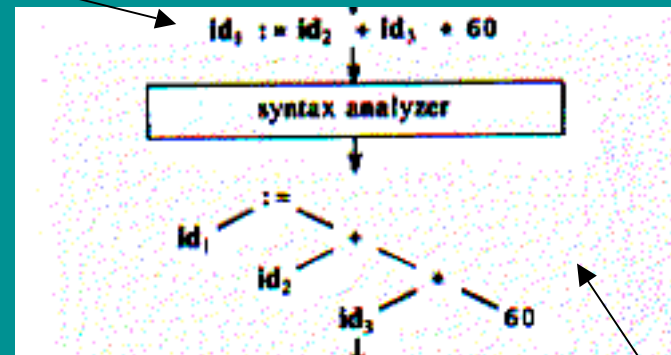
Esempio tratto da:

A.V. Aho, R. Sethi and J.D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1988

Stringa di simboli invece che stringa di caratteri

SYMBOL TABLE

| | | |
|---|----------|-----|
| 1 | position | ... |
| 2 | initial | ... |
| 3 | rate | ... |
| 4 | | |



Un albero (grafo, dag) ben rappresenta la struttura sintattica di una frase

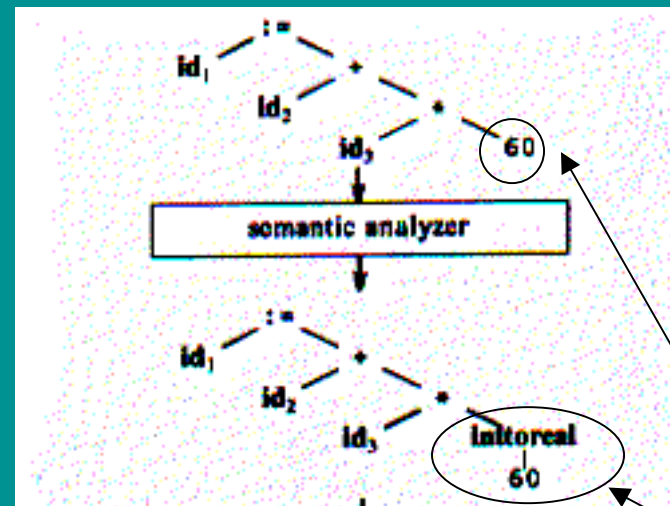
Esempio tratto da:

A.V. Aho, R. Sethi and J.D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1988

SYMBOL TABLE

| | | |
|---|----------|-----|
| 1 | position | ... |
| 2 | initial | ... |
| 3 | rate | ... |
| 4 | | |

Aggiungiamo le nuove informazioni raccolte nel visitare la struttura: tipi, uso degli identificatori, unita' di memoria da allocare, etc.



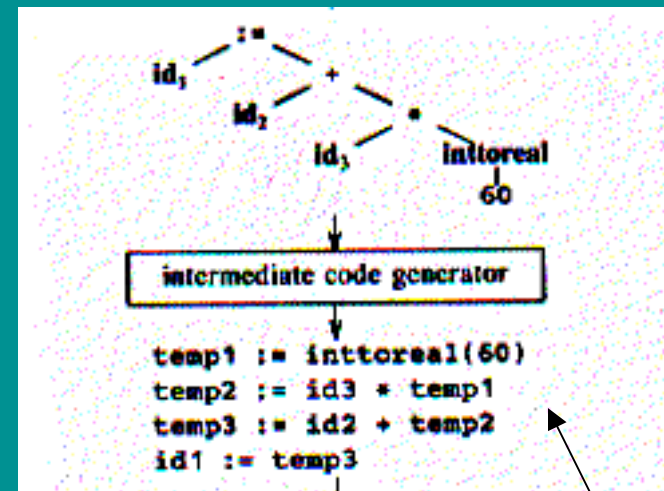
Si puo' modificare la frase quando si trovano incompatibilita' risolvibili

Esempio tratto da:

A.V. Aho, R. Sethi and J.D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1988

SYMBOL TABLE

| | | |
|---|----------|-----|
| 1 | position | ... |
| 2 | initial | ... |
| 3 | rate | ... |
| 4 | | |



Un linguaggio intermedio standard come il linguaggio a 3 indirizzi fornisce strumenti per l'ottimizzazione e la generazione di codice delle fasi successive

Useremo uno stesso strumento sia per l'analisi semantica sia per la generazione di codice (intermedio): Attribute Grammars

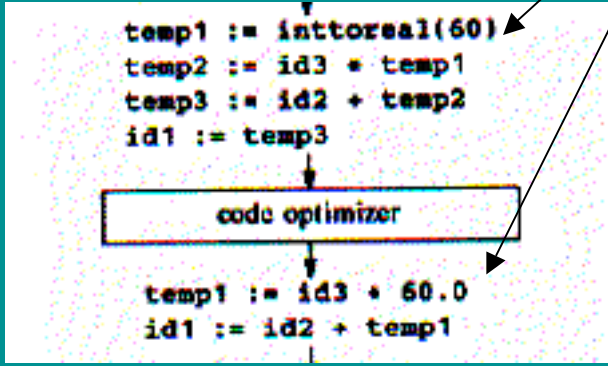
Esempio tratto da:

A.V. Aho, R. Sethi and J.D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1988

Al tempo di Compilazione possiamo risolvere alcune operazioni

SYMBOL TABLE

| | | |
|---|----------|-----|
| 1 | position | ... |
| 2 | initial | ... |
| 3 | rate | ... |
| 4 | | |



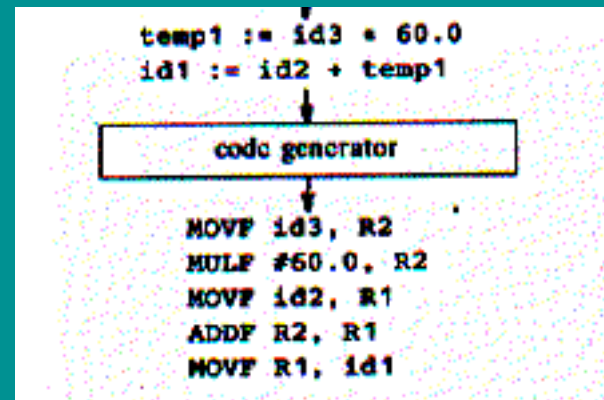
Conoscendo le proprietà del linguaggio intermedio
Possiamo individuare costruzioni inutili o ridondanti:
Qui eliminiamo l'impiego di una variabile dopo aver
Risolto una sottoespressione

Esempio tratto da:

A.V. Aho, R. Sethi and J.D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1988

SYMBOL TABLE

| | | |
|---|----------|-----|
| 1 | position | ... |
| 2 | initial | ... |
| 3 | rate | ... |
| 4 | | |



La generazione di codice dipende solo dal linguaggio intermedio e dal target: E' indipendente dal linguaggio sorgente