Nr.	Regel	Befehl	Kosten
1	Dreg → Areg	`move.l An, Dn`	4
2	Areg → Dreg	`move.l Dn, An`	4
3	Areg → @(Areg)	`move.l (An),An`	12
4	Dreg → @(Areg)	`move.l (An),Dn`	12
5	Dreg → +(Areg,Dreg)	`add.l An,Dn`	8
6	Dreg → +(Dreg,Dreg)	`add.l Dn,Dn`	8
7	Areg → +(Dreg,Areg)	`add.l Dn,An`	8
8	Areg → +(Areg,Areg)	`add.l An,An`	8
9	Dreg → +(@(Areg),Dreg)	`add.l (An),Dn`	14
10	Areg → +(@(Areg),Areg)	`add.l (An),An`	14
11	Dreg → $@_b$(Areg)	`move.b (An),Dn`	8
12	Dreg → $@_b$(+(Areg,Dreg))	`move.b 0(An,Dn),Dn`	14
13	Dreg → $@_b$(+(Areg,Areg))	`move.b 0(An,An),Dn`	14
14	Areg → `an`	ε	0
15	Dreg → `dn`	ε	0

Abbildung 8.2: Teil einer Befehlsauswahl-Baumgrammatik für den 68000

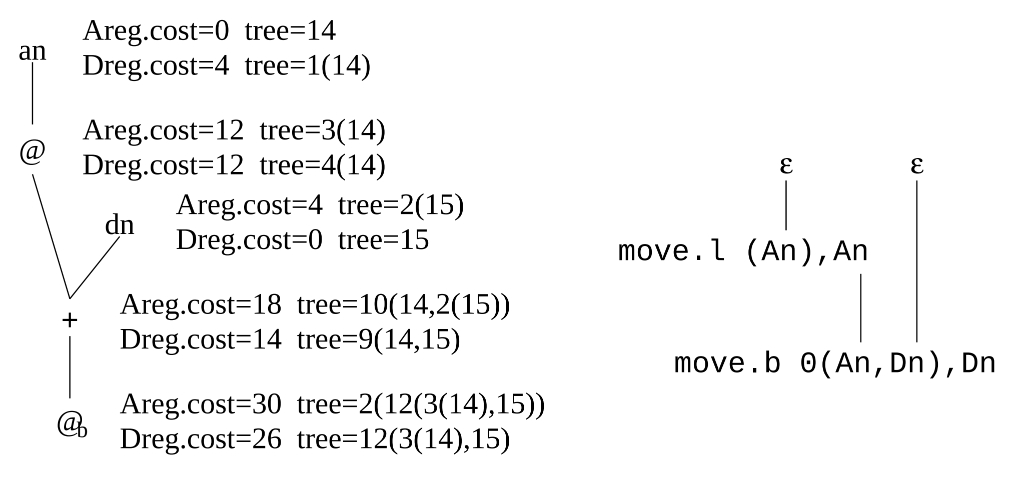

Abbildung 8.3: Optimale Befehlsauswahl

folgendermaßen: Die Regel verwendet den @-Knoten des Baums als Areg (Kosten: 12) und den dn-Knoten des Baums als Dreg (Kosten: 0), die Regel selbst kostet 14 Zyklen, die Gesamtkosten sind daher 26 Zyklen. Für Regel 11